ALEXANDRE SILVESTRE FERREIRA

A CROSS-DOMAIN MULTI-ARMED BANDIT HYPER-HEURISTIC

Dissertation presented in partial fulfillment of the requirements for the degree of Master in Informatics. Postgraduated program in Informatics, Sector of Exact Sciences, Federal University of Paraná.
Advisor: Prof. Dr. Aurora Pozo
Co-advisor: Prof. Dr. Richard Aderbal Gonçalves

CURITIBA

2016

ALEXANDRE SILVESTRE FERREIRA

A CROSS-DOMAIN MULTI-ARMED BANDIT HYPER-HEURISTIC

Dissertation presented in partial fulfillment of the requirements for the degree of Master in Informatics. Postgraduated program in Informatics, Sector of Exact Sciences, Federal University of Paraná.
Advisor: Prof. Dr. Aurora Pozo
Co-advisor: Prof. Dr. Richard Aderbal Gonçalves

CURITIBA

2016

F383c Ferreira, Alexandre Silvestre
A cross-domain multi-armed bandit hyper-heuristic/ Alexandre Silvestre Ferreira. – Curitiba, 2016.
70 f. : il. color. ; 30 cm.

Dissertação - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-graduação em Informática, 2016.

Orientador: Aurora Pozo – Co-orientador: Richard Aderbal Gonçalves.
Bibliografia: p. 64-70.

1. Algoritmos genéticos. 2. Algoritmo - Otimização. 3. Programação heurística. I. Universidade Federal do Paraná. II.Pozo, Aurora. III. Gonçalves, Richard Aderbal . IV. Título.

CDD: 006.3823

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
Setor CIÊNCIAS EXATAS
Programa de Pós Graduação em INFORMÁTICA
Código CAPES: 40001016034P5

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **ALEXANDRE SILVESTRE FERREIRA**, intitulada: "**A CROSS-DOMAIN MULTI-ARMED BANDIT HYPER-HEURISTIC**", após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO.

Curitiba, 26 de Fevereiro de 2016.

Prof AURORA TRINIDAD RAMIREZ POZO (UFPR)
(Presidente da Banca Examinadora)

Prof LUIZ EDUARDO SOARES DE OLIVEIRA (UFPR)

Prof ROBERTO SANTANA HERMIDA (UPV)

Prof RICHARD ADERBAL GONÇALVES (UNICENTRO)
Coorientador

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
Setor de Ciências Exatas
Coordenação de Pós-Graduação em Informática

CONTENTS

LIST OF FIGURES

LIST OF TABLES

PUBLICATIONS

Throughout the master degree, three papers were published:

The first work entitled "Aplicação do Algoritmo ACO-HH para o problema de cobertura de conjuntos" was published in the national conference Encontro Nacional de Inteligência Artificial e Computacional (ENIAC), 2014. It was an application of the an Ant Colony based selection hyper-heuristic for the Set Covering Problem. [3]. The approach showed promising results concerning large instances.

An extended version of the previous work was published in Advances in Distributed Computing and Artificial Intelligence Journal (ADCAIJ) [4]. A Statistical test was performed to compare the hyper-heuristic to other ant colony approach. The results confirmed that for three of the larger instances, there was no statistical difference in the quality of the results, but the hyper-heuristic had a much faster execution time.

Finally, a work entitled "A Multi-armed Bandit Hyper-heuristic", with the early results of the proposed approach, was published on the Brazilian Conference on Intelligent Systems, 2015 [In press]. The approach was tested in two domains Bin Packing and Personnel Scheduling and presented promising results.

RESUMO

Muitos problemas de otimização do mundo real são complexos e possuem muitas variáveis e restrições. Por esta causa, o uso de meta-heurísticas tornou-se a principal maneira de resolver problemas com essas características. Uma das principais desvantagens do uso de meta-heurísticas é que são geralmente desenvolvidas utilizando características do domínio fazendo com que sejam atreladas a ele dificultando sua utilização em outros problemas. Em buscas de algoritmos mais adaptáveis o conceito de hiper-heurísticas surgiu. Hiper-heurísticas são métodos de busca que visam solucionar problemas de otimização selecionando ou gerando heurísticas. Hiper-heurísticas de seleção escolhem uma boa heurística para ser aplicada a partir de um conjunto de heurísticas. O método de seleção é a principal peça de uma hiper-heurística de seleção tendo impacto fundamental em sua performance. Apesar de existirem vários trabalhos sobre hiper-heurísticas de seleção, ainda não existe consenso sobre como uma boa estratégia de seleção deve ser definida. Em busca de uma estratégia de seleção, algoritmos inspirados nos conceitos do problema Multi-Armed Bandit (MAB) serão estudados. Estes algoritmos foram aplicados ao contexto da Seleção Adaptativa de Operadores obtendo resultados promissores. Entretanto, ainda existem poucas abordagens para o contexto de hiper-heurísticas.

Nesta dissertação propomos uma hiper-heurística que utiliza algoritmos MAB como sua estratégia de seleção. A abordagem proposta é desenvolvida utilizando o *framework* HyFlex, que foi proposto para facilitar a implementação e comparação de novas Hiper-heurísticas. Os parâmetros foram configurados através de um estudo empírico, e a melhor configuração encontrada foi comparada com os 10 primeiros colocados da competição CHeSC 2011. Os resultados obtidos foram bons e comparáveis com os das melhores abordagens da literatura. O algoritmo proposto alcançou a quarta colocação. Apesar dos bons resultados, os experimentos demonstram que a abordagem proposta sofre grande influência dos parâmetros. Trabalhos futuros irão investigar formas de amenizar esta influência.

ABSTRACT

Many real word optimization problems are very complex with many variables and constraints, and cannot be solved by exact methods in a reasonable computational time. As an alternative, meta-heuristics emerged as an efficient way to solve this type of problems even though they cannot ensure optimal values. The main issue of meta-heuristics is that they are built using domain-specific knowledge, therefore they require a great effort to be used in a new domain. In order to solve this problem, the concept of Hyper-heuristics were proposed. Hyper-heuristics are search methods that aim to solve optimization problems by selecting or generating heuristics. Selection hyper-heuristics choose from a pool of heuristics a good one to be applied at the current stage of the optimization process. The selection mechanism is the main part of a selection hyper-heuristic and has a great impact on its performance. Although there are several works focused on selection hyper-heuristics, there is no unanimity about which is the best way to define a selection strategy. In this dissertation, a deterministic selection strategy based on the concepts of the Multi-Armed Bandit (MAB) problem is proposed to cross-domain optimization. Multi-armed bandit approaches define a selection function with two components, the first is based on the performance of an operator and the second based on the number of times that the operator was used. These approaches had showed a promising performance over the Adaptive Operator Selection context. However, there are few works on literature that aim the hyper-heuristic context, as proposed here. The proposed approach is integrated into the HyFlex framework, that was developed to facilitate the implementation and comparison of hyper-heuristics. An empirical parameter configuration was performed and the best setup was compared to the top ten CHeSC 2011 algorithms using the same methodology adopted during the competition. The results obtained were good comparable to those attained by the literature. Moreover, it was concluded that the behavior of MAB selection is heavily affected by its parameters. As this is not a desirable behavior to hyper-heuristics, future research will investigate ways to better deal with the parameter setting.

CHAPTER 1

INTRODUCTION

Many optimization problems that are found on real world are very complex and have many constraints, for that reason most of them cannot be solved using exact methods in a reasonable computational time [5]. As a result, the use of metaheuristics approaches, that do not have any guarantee of returning the best solution but usually returns a good solution, became not only a trend but also one of the main tools to solve optimization problems.

Metaheuristics had proven to be an efficient method and were successful applied on several problem domains. However, they have the issue of being built upon domain-specific knowledge requiring a great effort from the practitioner to determine: how the domain will be modeled, which are the best heuristics to be used, which is the best set of parameters, etc. Each one of those characteristics has great influence over the performance of a metaheuristic. Therefore, the best algorithms usually are those expertly crafted by incorporating characteristics of the chosen problem instance [1, 6]. The use of problem knowledge makes it very difficult to generalize or apply meta-heuristics over different/new problems, causing metaheuristics to be re-developed in order to solve problems from a different domain [7].

Thus, the automated heuristic design has emerged as an efficient way to enhance search algorithms by adjusting parameters and operators in an on-line way [7], one example of this methodology are Hyper-heuristics [8].

Hyper-heuristics are search methods that solve optimization problems by exploring the search space of a given set of heuristics rather than the solution space directly and one of its main concepts is to be able of solving multiple problems (cross-domain) by being general. They can be classified into two categories: selection and generation. Hyper-heuristics from the first category selects a good heuristic from a pool of heuristics to be applied at each step while hyper-heuristics from the second category build heuristics from a pool of heuristics' components.

This work focuses on selection hyper-heuristics.

1.1 Motivation and Goals

Selection hyper-heuristics are the most used type of hyper-heuristic on literature due its simplicity and efficiency. Although selection hyper-heuristics are generally robust, two tasks have a great impact on their performance: the heuristic selection and the move acceptance mechanism [9]. Even with several works about selection hyper-heuristics, what does a selection strategy to perform well, which is the best selection strategy and which move acceptance should be used are still open research questions.

The early approaches of hyper-heuristics focused on just one domain and were usually applied on scheduling problems where they achieved good results overcoming state-of-art approaches in some instances [10, 11, 12, 13]. However, the performance in other domains was not so good [14, 15, 16]. The interest about the application on several domains grew in the year of 2011 with the organization of the first Cross-domain Heuristic Search Challenge (CHeSC), by the Automated Scheduling, Optimization and Planning (ASAP) group at the University of Nottingham, with the objective to encourage the research about cross-domain hyper-heuristics. In order to standardize the implementations and facilitate the comparison, a framework called HyFlex was proposed [17] including definitions and heuristics to six problem domains: Boolean Satisfiability [18], Bin Packing [19], Personnel Scheduling [20], Permutation Flow Shop [21], Traveling Salesman [22] and Vehicle Routing [23]. As soon as the CHeSC Challenge was over, it became a benchmark for new selection hyper-heuristics due to the great number of algorithms proposed.

Therefore, in this dissertation a selection strategy based on Multi-armed Bandit (MAB) is proposed. Multi-armed bandit algorithms have been successfully applied on the Adaptive Operator Selection (AOS) context, achieving good results [24]. As the task of selecting a heuristic is very similar to the selection of an operator for the AOS context, the algorithms that are used on one context can be used on the other. Another motivation is that there are few works that utilize MAB algorithms with hyper-heuristics on literature, most of them for multi-objective optimization applied to one or two domains.

Thus, the main goal of this dissertation is to propose and analyze the behavior of a selection hyper-heuristic with MAB based selection strategy to cross-domain mono-objective optimization. In order to do that, three algorithms were chosen to perform the selection task, they were: Sliding Window Multi-armed Bandit (SLMAB), Fitness-Rate-Rank-based Multi-armed Bandit (FRRMAB) and Fitness-Rate-Average Multi-armed Bandit (FRAMAB). The first two algorithms were proposed by Fialho on [24] and Li [25] for AOS context. The last, is a mixture of FRRMAB with some components of SLMAB and is proposed here. These selection methods were combined with nine acceptance mechanisms, the majority of them classical, in order to define a good one for these new selection strategies.

To evaluate and analyze the approach, an empirical experiment was conducted using the methodology of CHeSC 2011 competition [2]. The approach was compared with the top ten approaches from CHeSC benchmark and a recent Dynamic Multi-armed Bandit approach proposed in [26].

1.2 Structure of the Dissertation

The dissertation is structured as follows:

Chapter 2 This chapter presents the related works used for this dissertation.

Chapter 3 In this chapter the main concepts of Hyper-heuristics are presented. First the definition of hyper-heuristics is presented followed by its classifications then a more in-depth view about selection hyper-heuristics and its components: the selection method and the move acceptance.

Chapter 4 This chapter presents the concepts of Multi-armed Bandit algorithms and its details, it also presents a new variation proposed in this work and the credit assignment schemes used.

Chapter 5 This chapter presents the details about the proposed hyper-heuristic.

Chapter 6 This chapter presents details about the experimental phase of the work, the methodology, the problem domains used, and how the parameters were configured. It also presents the results and the comparison made. Finally, a discussion about the algorithm performance is also presented.

Chapter 7 This chapter provides a general summary of the dissertation as well as the final thoughts about the results and directions for future research.

CHAPTER 2

RELATED WORKS

This Chapter presents the works related to this proposal. These works were selected based on its relation to the approach proposed in this dissertation and are divided into two parts. Initially, the works used in the development of this approach are presented. Those works are based on the Upper Confidence Bound (UCB) algorithm [27] and applied to the Adaptive Operator Selection (AOS) context. An in-depth review of UCB algorithms and Adaptive Operator Selection can be found in [28]. The second part focuses on cross-domain selection hyper-heuristic approaches, an in-depth review about hyper-heuristics can be found in [8].

In [24] several strategies for the Adaptive Operator Selection based on the UCB algorithm and credit assignment schemes were proposed.

The Multi-armed Bandit Algorithm (MAB) was proposed in [24], it was a straightforward implementation of UCB, with the addition of a scaling factor parameter in order to deal with the different scales values of different problems. The algorithm was tested on several domains and achieved good results on uniform instances. On the other hand, on instances that need a fast adaptation, the results presented a large decrease on its values. As this algorithm stores the improvements obtained for all operators from the begging of the search, the performance values accumulate, making it difficult for the algorithm to detect a new promising operator.

The Dynamic Multi-armed Bandit (DMAB) was first proposed in [29] and adapted to the AOS in [24]. It is a combination of the MAB algorithm with the Page-Hinkley (PH) statistical test, in order to have an adaptable algorithm. This algorithm performs statistical tests over the operators to check if the current best operator still is the best, if the answer is no, then the algorithm restart from scratch. This is done in order to rapidly adapt to the new context. In [30], this method was compared to state-of-art approaches of Adaptive Operator Selection and its results outperformed the other methods. In [24] DMAB was compared to several other approaches on many domains and achieved good results. The main drawback of this approach was that, in order to achieve those results, it had to be carefully tuned for each domain.

The Sliding Window Multi-armed Bandit (SLMAB), was proposed in [24] and introduced the idea of using a sliding window structure in order to better adapt to the dynamic context of Adaptive Operator selection. The idea is to use a sliding window with fixed size and a First In First Out policy, to store the performances of all operators. The goal is to use only the recent results to select the operator to be applied. Another concept introduced in this algorithm is that, to accelerate the identification of promising operators, the reward received by an operator that was applied few times should be higher then the reward received by an operator that is being constantly applied. The algorithm was compared to DMAB and MAB, the results showed that it was inferior to DMAB in most of the cases, and lost to MAB in some domains. It was noticed that the parameters have a great influence on its performance.

The Ranked-based Multi-armed Bandit (RMAB) was another approach proposed in [24]. The great sensitivity of the previous approaches to the parameters was the motivation to develop a more robust approach. The robustness is achieved by using a rank credit assignment with normalized values. To perform the exploration, the update of the performance values of all operators after is done at the end of an iteration. Although, the great influence of the parameters on the performance was still noticed, the results showed that this approach is more robust achieving comparable values to other approaches in several domains, without great variance of the parameters values.

In [25], the Fitness-Rank-Rate-based Multi-armed Bandit (FRRMAB) was proposed for Adaptive Operator Selection on a multi-objective algorithm MOEA/D. This approach uses the performance values of the sliding window (implemented as on SLMAB) to create a rank that is used to give credit to the operators, it also uses a decay factor that can be used to increase the rewards of the best-ranked components. An interesting point is that, differently of the other MAB approaches presented here, the value received by the credit scheme is directly defined as its performance, giving a more dynamical behavior. The results indicate that the AOS with FRRMAB can significantly increase the performance of an MOEA/D. Based on FRRMAB, in [31] a hyper-heuristic approach for multi-objective context was proposed. Two variations were proposed one based on the UCB and the other on UCB-Tunned, a modification of UCB that includes the variation of the performances [27]. In that work, the MOEA/D with UCB-Tuned produced favorable results when compared with the other two MAB-based methods and other two state-of-art Adaptive Operator Selection MOEA/D.

Although one of the concepts of hyper-heuristics is to be applied to several domains without great modifications, most of the early works proposed hyper-heuristics to one or two domains [8]. With the growing interest in the cross-domain application, on the year of 2011, the first CHeSC Challenge was created [2] and several approaches were proposed since then. These works are presented below.

In [32] a hyper-heuristics based on the meta-heuristic Ant Colony Optimization called ACO-HH was proposed. Each ant builds a path of hyper-heuristics, the pheromone updated is done by the average of improvement obtained by all ants and the heuristic information is the average of improvement obtained by specific movements of the low-level heuristics. This approach achieved the 11th place on the challenge. In [?] we proposed the ACO-HH applied to the Set Covering Problem (SCP), the results showed comparable results for the big instances and a great improvement on the execution time.

In [33] an approach called KSATS-HH that combines reinforcement learning, Tabu Search, and Simulated Annealing was proposed. The heuristic selection is done according to a tournament based on the rank value of the heuristics, this rank is incremented by one if the heuristic achieved an improvement and decreased otherwise. If a heuristic did not achieve any improvement it will be put on a Tabu list for seven iterations. In order to accept the solutions generated a Simulated Annealing move acceptance mechanism is used. This approach achieved the 9th place on the challenge.

In [34] the Variable neighborhood search based hyper-heuristic VNS-TW was proposed. It consists of four steps: shaking, local search, environmental selection, and periodical adjustment. In the first step, a low-level heuristic from mutation or ruin-recreate heuristic is applied. Then, a local search low-level heuristic is applied. A tournament selection is used to determine which solution of the population will be replaced and at the end, the parameters of the low-level heuristics are updated. This approach achieved the 3th place on the challenge.

In [35] a hyper-heuristic called ADAP-HH was proposed, its main goal was to be as general as possible. Several mechanisms of this approach worth a mention, a low-level heuristic exclusion mechanism was developed with a Tabu list in order to remove bad-performing heuristic from the selection list. An additional exclusion is performed related to execution time. For the acceptance, a new mechanism based on threshold was developed, this mechanism reset the solution if there were several iterations without improvement. Finally, a reinforcement learning technique is used to update the parameters of the low-level heuristics based on its previous performance. This approach achieved the 1st place on the challenge.

A choice function approach was proposed in [36]. This approach used a reinforcement learning mechanism in order to update the parameters of the choice function. The acceptance mechanism used was All Moves and the approach achieved the 12th place.

Recently, two approaches had overcome the results of ADAP-HH, they were [37] and [7]. The first one utilizes a Hidden Markov Model in order to define good sequences of heuristics and acceptances. The second utilizes a gene expression programming to create the selection strategy and the move acceptance.

The same authors from [7], proposed a DMAB approach for cross-domain in [26]. The DMAB is used as the selection strategy and combined with move acceptances generated by gene expression programming. This approach achieved the 4th place and will be used for comparison in this dissertation.

CHAPTER 3

HYPER-HEURISTICS

Hyper-heuristics have emerged as an important research field on the Optimization area and can be defined as search methods that aim to solve optimization problems by selecting or generating heuristics [1].

Their motivation came from the No Free Lunch Theorem which establish that "for any algorithm, any elevated performance over one class of problems is a offset by diminished performance over another class" [38]. This theorem applies to Meta-heuristics, since in order to find good solutions those methods often need to be designed and tuned taking into account the problem domain or even just a single problem instance. Hyper-heuristics try to overcome this by increasing its generality.

On a Hyper-heuristics two types of heuristics are defined, the high and the low-level ones. Regarding those levels, high-level heuristics are responsible for choosing which low-level heuristic to apply/generate and which solutions generated will be accepted (replacing one or more of them). Low-level heuristics are responsible for solving the problem (searching through the solution space). High-level heuristics are, generally, independent from the problem while low-level heuristics are problem dependent. Thus, a hyper-heuristic operates upon the low-level heuristics search space rather than the solution space creating a more adaptable algorithm [1].

This chapter presents a classification of hyper-heuristics, a selection framework and some details about the selection and move acceptance mechanisms.

3.1 Classification

According to Burke et al. [1], hyper-heuristics can be classified as selection hyper-heuristics or generation hyper-heuristics based on how they use the low-level heuristics.

Hyper-heuristics can also be classified based on the learning paradigm employed as on-line, off-line or no learning. In on-line hyper-heuristics, the learning process occurs during the optimization of the problem while in off-line hyper-heuristics a set of benchmark instances is used for the learning process. No learning hyper-heuristics use a fixed rule to select or generate low-level heuristics [1].

Another characteristic of hyper-heuristics is the type of low-level heuristics that is selected/generated which can be constructive or perturbative. Constructive heuristics build a solution by selecting a component at each search step; while perturbative heuristics transform a complete solution into another complete solution, i.e., constructive heuristics accept and generate partial solutions while perturbative heuristics allow only complete solutions [1]. Figure 3.1 presents a representation of this classification.

Most of the approaches found in literature are selection hyper-heuristics with on-line learning capabilities that operates over the perturbative low-level heuristics, and that is also the case of the present approach.

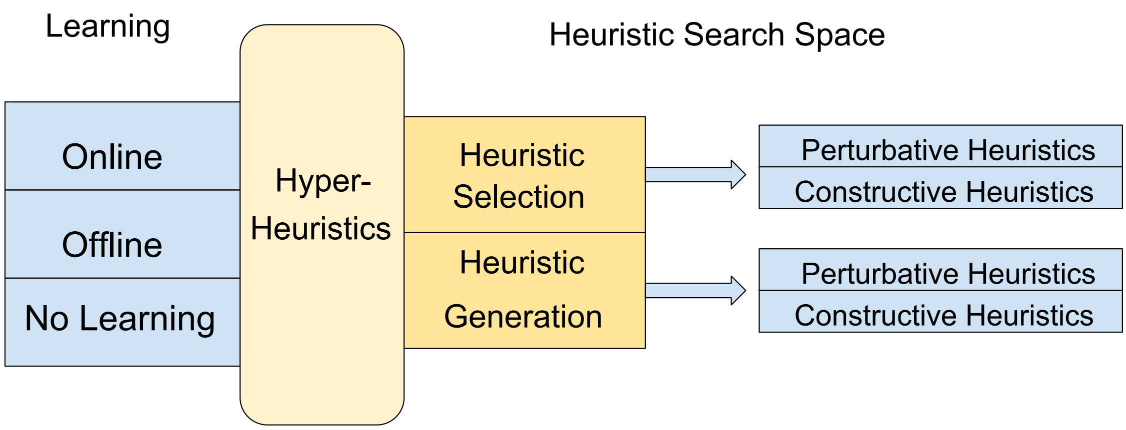

Figure 3.1: Classification of Hyper-heuristics adapted from [1].

3.2 General Hyper-heuristic Framework

Cowling, Kendall and Soibeiga in [39], proposed that a hyper-heuristic framework should have two levels:

1. Hyper-heuristic: composed by the high-level heuristics and other functions that are independent of the problem.

2. Problem Domain: composed by the low-level heuristics, the evaluation function and any other structure or operator which need information about the problem.

In order to block the exchange of unnecessary information between the two levels there is a conceptual barrier called "Domain Barrier". Thus, the high-level heuristic (the core of the hyper-heuristic) only has knowledge about the number of heuristics, their performance values and other problem independent data [40]. Therefore, if a hyper-heuristic follows this framework it can be employed on different problem domains without any modification on the high-level. A representation of this framework is presented on Figure 3.2

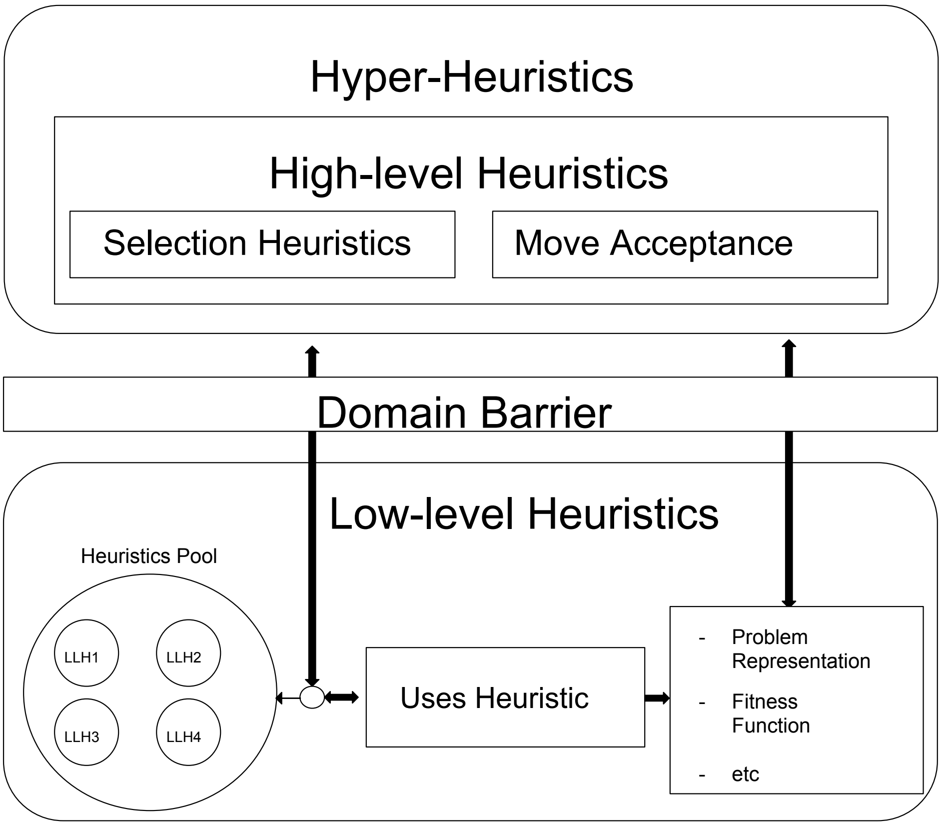

Figure 3.2: A Hyper-heuristic Framework.

3.3 Selection Hyper-heuristics

Selection hyper-heuristics aims to select the most promising low-level heuristic at the current search state in order to solve the problem. Briefly, a selection hyper-heuristic operates as follows: a high-level heuristic selects one of the low-level heuristics based on some measurement of the performance (i.e., solution improvement). Then, the selected low-level heuristic is applied to the current solution and the resultant performance is returned to the hyper-heuristic that uses an acceptance mechanism to decide if the new solution is accepted or not (move acceptance).

3.3.1 Selection Strategies

Regarding the selection strategies, its functions is to determine a strategy for choosing the most desirable heuristics during different phases of the search process, as stated before this mechanism only has knowledge of problem independent information (for example, the improvement obtained by the application of a heuristic, the number of low-level heuristics and the execution time).

Although it might seem simple, the selection task is an instance of the exploitation versus exploration (EvE) dilemma. On one hand it wants to select the best performing heuristic as many times as possible (exploitation), on the other hand it should verify if there is another heuristic that is better than the current one (exploration). Another important point is what information is used in order to evaluate the quality of the low-level heuristics, although the function improvement is a good indicator, the use of a raw value turns the algorithm too sensitive to the scales adopted by the different domains or even between different instances of the same domain. Hence, the use of ranked or normalized values are the common choice.

The most popular selection strategy adopted by hyper-heuristics is the Choice Function (CF), which was proposed in [39]. It is a function that chooses which low-level heuristic to apply based on the CF value that is calculated using three components: (i) the improvement obtained by the low-level heuristic, (ii) the improvement obtained by the combination of two low-level heuristics and (iii) the time passed since the heuristic was called the last time. Components (i) and (ii) favors exploitation while (iii) favors exploration. The CF has been proved to be efficient and was used on several works [41, 36, 42, 43].

The use of meta-heuristics as the selection strategy or an embedded function on hyper-heuristics is very common, although the results are usually good one drawback by those approaches is that the meta-heuristics parameters must be tuned as on a classic implementation decreasing the hyper-heuristics generality.

The use of evolutionary meta-heuristics as a selection mechanism on hyper-heuristics can be found [44, 45, 46, 47], while [48, 49, **?**] present examples of ant colony optimization selection strategies.

Recently there is a growing interest on Multi-Armed Bandit algorithms (see Chapter 4). Multi-Armed Bandit based selection strategies are one of the focuses of this dissertation.

3.3.2 Move Acceptance

The acceptance mechanism of a hyper-heuristic determines the way that the search space will be explored and has a major influence over a hyper-heuristic performance [9]. Although a policy that accepts only better solutions seems intuitively good, it generally leads to a local optimal. That being said, the acceptance of worsen solutions is necessary, but on a controlled manner. The following acceptance mechanisms are considered in this work (the definitions presented below are for minimization but could be changed to maximization):

- All Moves (AM): All solutions created are accepted without consideration of their quality.
- Improving or Equal(IE): The generated solution is accepted if the objective value is equal or better than the previous one. Worsen solutions are accepted if a number d of iterations without improvement is reached. In this work $d = 120$ as in [50].
- Only Improving (OI): Improving solutions are always accepted, no improving solutions are never accepted.
- Naive Acceptance(NA): Improving or equal solutions are always accepted. Non improving solutions are accepted with 50% probability [51].

- Simulated Annealing (SA): Improving or Equal solutions are always accepted. However, non-improving solutions are accepted based on a probability acceptance function , $R < exp(-\delta/t)$, where R is a random number in the interval [0,1] and δ is the change in the objective value. The ratio of accepted non-improving solutions is controlled by a temperature t which gradually decreases by β during the search process. In this work $t = 1$ and β is set accordingly to the time taken to execute 15 iterations of the proposed heuristic as in [33].

- Exponential Monte Carlo (EMC): Improving or equal solutions are always accepted. Worsen solutions are accepted with a probability of $R < exp(-\delta)$, where R is a random number between [0,1] and δ is the change in the function value. The ratio of acceptance decreases as δ increases [52].

- Great Deluge (GD): Improving or equal solutions are always accepted. Worsen solutions are accepted if its objective value is less than a *threshold* initially set as the value of the initial solution. The value of the *threshold* is gradually decreased by β. The value of β is calculated by Equation 3.1

$$\beta = \frac{f(initial_{solution})}{t} \tag{3.1}$$

  where t is the time taken by the hyper-heuristics to execute 15 iterations (this values came from [33]).

- Record to Record (RR): Improving or equal solutions are always accepted. Worsen solutions are accepted if the objective value is less than $R + D$, where R is the value of the initial solution and D a deviation. In this work as in [53], $D = 0.03$ and R is updated every iteration, being modified to the value associated with the current solution.

- Adaptive Acceptance(AA): Improving or equal solutions are always accepted. Worsen solutions are accepted according to an acceptance rate AR which is updated during the search. Initially, AR is set to zero. However if the solutions cannot be improved for a certain number of iterations n, the AR value is increased by 5%. For this work whenever a solutions is accepted AR goes back to zero [51].

3.4 Summary

In this chapter were presented the main concepts of hyper-heuristics, how they are classified and its general framework. A more in-depth view about selection hyper-heuristics defining how they work and their two components: the selection strategy and the move acceptance mechanisms were also presented. For the move acceptance, nine approaches that will be used with the proposed approach were presented.

Regarding the selection strategy, this dissertation will utilize a Multi-armed Bandit based approach. The concepts and definitions about Multi-armed Bandit problems and the algorithms used in the selection strategy will be presented in the next chapter.

CHAPTER 4

MULTI-ARMED BANDIT

Multi-armed bandit problems (MAB) were introduced by [54] and have been the focus of several studies by the Statistical community as it offers a very clean and simple theoretical formulation, for analyzing the exploration and exploitation (EvE) dilemma [28].

In its statistical formulation, a bandit problem consists of set of K probability distributions $\langle p_1, ..., p_k \rangle$ with expected values $\langle \mu_1, ..., \mu_k \rangle$ and variances $\langle \sigma_1, ..., \sigma_k \rangle$ the goal is to obtain the maximum reward by selecting probabilities p_i with the best returning value. As the probabilities and values are initially unknown to the player, they can be interpreted as corresponding arms on a slot machine (Figure 4.1). Therefore, the player can be seen as a gambler whose goal is to collect as much money as possible by pulling the arms over several turns. At each turn t, the player selects an arm $j(t)$, and receives a reward $r(t) \sim P_{j(t)}$. For now on, the player will be seen as the hyper-heuristic, the arms as low-level heuristics and the money as the low-level heuristics performance. Therefore, bandit algorithms specify a strategy to determine which heuristic should be selected by the hyper-heuristic on each turn.

Figure 4.1: A bandit machine.

Many efficient ways to solve the MAB problem were proposed in literature [28], the Upper Confidence Bound (UCB-1) algorithm [27] is a method that ensures asymptotic optimality in terms of cumulative reward. An UCB-1 strategy is based on two components:

the first one is related to the performance of the heuristic and the second is related to the number of times that the heuristic was applied. Thus, the selection of a heuristic h on the turn t is given by Equation 4.1.

$$h(t) = \ argmax_{h=1..K} \left(q_{h,t} + \sqrt{\frac{2log \sum_{i=0}^{K} n_{i,t}}{n_{h,t}}} \right) \tag{4.1}$$

Where the first component $q_{h,t}$ represents the quality of the h-th heuristic and is related to exploitation, while the second one gives an upper confidence bound based on the number of times $n_{h,t}$ that the heuristic was selected and is related to exploration. K is the number of heuristics in the pool.

In [24], Fialho proposed four algorithms based on the UCB that produced good results in the Adaptive Operator Selection Context, in order to handle with the difference between the rewards scales he introduced a hyper-parameter C to the UCB selection equation as shown in 4.2.

$$h(t) = \ argmax_{h=1..K} \left(q_{h,t} + C \cdot \sqrt{\frac{2log \sum_{i=0}^{K} n_{i,t}}{n_{h,t}}} \right) \tag{4.2}$$

All the strategies presented bellow use the same selection equation and differ by how the two components values $q_{h,t}$ and $n_{h,t}$ are defined. A credit assignment scheme is used to define the reward $r_{h,t}$ of a heuristic, this reward is used to calculate $q_{h,t}$.

4.1 Multi-armed Bandit Algorithm

The MAB algorithm is a straightforward implementation of the UCB, it defines the quality of a heuristic $q_{h,t}$ as the average of rewards $r_{h,t}$ obtained by the heuristic and $n_{h,t}$ as the number of times that the heuristic was applied. Those values are updated as shown in equation 4.3 and 4.4.

$$n_{h,t+1} = n_{h,t} + 1 \tag{4.3}$$

$$q_{h,t+1} = \left(\frac{q_{h,t} + r_{h,t}}{n_{h,t}} \right) \tag{4.4}$$

One problem with this approach is that the performance of a particular heuristic can vary during the search process. As the quality $q_{h,t}$ is an average of all rewards obtained, this value may not reflect the current state of the performance of the low-level heuristics and this is not a desirable behavior in the selection context since it makes the algorithm less adaptable.

4.2 Dynamic Multi-armed Bandit

The DMAB algorithm is a hybridization of the original MAB algorithm (UCB with Scaling factor) with the PH statistical test [24]. It was proposed in [29] and adapted to AOS by [30].

The statistical test is used to detect changes in average reward of the current best low-level heuristics. After, if the current best low-level heuristic is not the best anymore, DMAB restarts from scratch [26]. The idea is to identify the new best low-level heuristic as soon as possible. Let σ_t represent the average of a heuristic over the last t time steps, $r_{h,j}$ represents the reward of the h-th heuristic at time j, the PH test detected the changes in average reward according to Equation 4.5.

$$\sigma_t = \frac{\sum_{j=1}^{t} r_{h,j}}{n_{h,t}}, \quad e_t = (r_{h,t} - \sigma_t + \gamma), \quad m_t = \sum_{j=1}^{t} e_j \tag{4.5}$$

where e_t is the difference between the current reward and the average reward of heuristic h plus a tolerance parameter γ. m_t is used to accumulate the differences until time t. The statistical difference is recognized when the difference is greater than a predefined threshold τ ($max_{j=1..t} \{|m_j|\} - |m_t| > \tau$) [26].

In combination with this method, the extreme of improvements was used, see Section 4.6.

4.3 Sliding Window Multi-armed Bandit

In order to make the MAB algorithm more adaptable, the SLMAB was proposed. It defines a sliding window with size W and a First in First Out (FIFO) insertion policy to store all heuristics rewards. Therefore only the W most recent rewards are considered to update the quality value. A well-adjusted sliding window is able of both: store a large amount of rewards in order to allow the correct choice of the best low-level heuristic and be sufficiently small to cancel or limit the fluctuation of the rewards.

Another feature of the SLMAB is that the reward of a heuristic that has been applied few times has a higher influence on its quality than the reward of a heuristic that has been applied many times, this is done to rapidly identify a promising heuristic. Considering the exploitation and exploration terms ($q_{h,t}$ and $n_{h,t}$) the update is done as shown in Equation 4.6.

$$\begin{cases} q_{h,t+1} = q_{h,t} \cdot \dfrac{W}{W + (t - t_h)} + r_{h,t} \cdot \dfrac{1}{n_{h,t} + 1} \\ n_{h,t+1} = n_{h,t} \cdot \left(\dfrac{W}{W + (t - t_h)} + \dfrac{1}{n_{h,t} + 1} \right) \end{cases} \tag{4.6}$$

Where t_h is the last time that the heuristic h was applied.

4.4 Fitness-Rate-Rank-based Multi-armed Bandit

The FRRMAB was the last MAB algorithm proposed by Fialho [25] and was designed paying attention to the dynamic nature of the adaptive operator selection problem. In order to better adapt to the changes the quality estimate of a low-level heuristic is defined as its reward as shown in Equation 4.7. A sliding window structure SW as the one presented on SLMAB is also used.

$$q_{h,t} = r_{h,t} \tag{4.7}$$

The second component $n_{h,t}$ is defined as the number of times that a heuristic appears in the sliding window, see Equation 4.8.

$$n_{h,t} = \sum_{i=1}^{W} X_h, \quad X_h = \begin{cases} 1, & if\ h = i \\ 0, & otherwise \end{cases} \tag{4.8}$$

The credit assignment presented with this approach is the Ranked Fitness Improvement and is discussed on Section 4.6.

4.5 Fitness-Rate-Average-based Multi-armed Bandit (FRAMAB)

The FRAMAB is the proposed selection method in this dissertation its idea came from a combination of some characteristics from SLMAB and FRRMAB. The quality measurement uses the reward obtained by the credit assignment but multiplies it by the inverse of the number of times that the low-level heuristic is on the window Equation 4.9.

$$q_{h,t} = r_{h,t} \cdot \frac{1}{n_{h,t}} \tag{4.9}$$

Where $n_{h,t}$ is defined as the number of times that a heuristic appears in the sliding window, see Equation 4.8.

$$n_{h,t} = \sum_{i=1}^{W} X_h, \quad X_h = \begin{cases} 1, & if\ h = i \\ 0, & otherwise \end{cases} \tag{4.10}$$

The quality measurement presented in Equation 4.9 gives both, an instant reward as the FRRMAB, but also adds a higher value to those heuristics that were applied few times, this is done in order to accelerate the identification of a promising heuristics. In combination with this method, the average of improvements was used, see Section 4.6.

4.6 Credit Assignment

The use of raw improvements in order to evaluate the quality of a component has proved to not be a good method for the selection context [24, 25], mainly because of the different scales and the great fluctuation that those values can have. Therefore, a more robust way to deal with those values is necessary, on the MAB context this is done through a Credit Assignment Scheme that can be defined in different ways.

According to [24], a credit assignment is defined by three aspects (i) how to measure the impact of an operator application; (ii) how to assign credit to the operator based on this measured impact; and (iii) to which operator the credit should be assigned to. It is important to notice that for all of them only the selected component (heuristic) receives the credit and, by convention, there is no negative credit. Let δ be the improvement obtained by one application of a low-level heuristic at time t, it is calculated as shown in Equation 4.11.

$$\delta_{h,t} = max\left(0, \frac{f(s)_{t-1} - f(s)_t}{f(s)_{t-1}}\right) \tag{4.11}$$

Where $f(s)_t$ is the function value associated to a solution s on time t. An important feature of this equation is the normalization embedded into the calculation, that reduces the impact of different function scales associated to different problem domains and facilitates the setting of the parameters as it does not need to deal with the different scales.

The Schemes considered in this dissertation will be presented next.

4.6.1 Instantaneous and Average Credit Assignment

The *Instantaneous* method gives credit to the latest improvement obtained (see Equation 4.12), because of that it tends to be a very unstable method due the fluctuations of the improvement values. A more robust and common approach is to use the average of the last W improvements, this method is called *Average* and can be mathematically defined as shown in Equation 4.13, where t is a time where the heuristic h was chosen.

$$r_{h,t} = \delta_{h,t} \tag{4.12}$$

$$r_{h,t} = \frac{\sum_{i=1}^{W} \delta_{h,i}}{n_{h,t}} \tag{4.13}$$

4.6.2 Extreme Credit Assignment

The Extreme Credit Assignment was proposed in [55], given two heuristics one bringing frequent, small improvements and compare it to an operator bringing rare large improvements, using the average credit assignment the last one will hardly be considered since its average is likely to be closer to 0 even though it might be the current best heuristic. In order to overcome that, the Extreme value-based Credit Assignment mechanism proceeds as follows [24].

Let t be the current time step and $t_{h,i}$ is a time where the heuristic h was chosen. Since $\delta(t)$ defines the improvement observed at time t, then the credit assigned to a heuristic h is computed by Equation 4.14.

$$r_{h,t} = max_{i=1...W}\left\{\delta_{h,i}\right\} \tag{4.14}$$

The pseudocode of Instantaneous, Average, and Extreme credit assignment schemes is given on Algorithm 1.

Algorithm 1: Credit assignment Schemes

```
Input: heuristic, type, W
if type == Instantaneous then
    r_h ← last(rewardsW_h)
else if type == Average then
    r_h ← avg(rewardsW_h)
else if type == Extreme then
    r_h ← max(rewardsW_h)
end
return r_h
```

4.6.3 Ranked Fitness Improvement

In order to give more robustness to the credit assignment, the use of ranks instead of selection schemes over raw fitness values was proposed [24]. In [25], all δ values from a heuristic h are summed as $Reward_h$ (Equation 4.15), then this value is ranked in descending order. Let $Rank_h$ be the rank value of a heuristic h, to give more chances to the best heuristics a decaying factor $D \in [0, 1]$ is used to modify the $Reward_h$ as seen in Equation

4.16, the smaller the D value, bigger is the influence of the best heuristic.

$$Reward_h = \sum_{i=1}^{W} \delta_{h,i} \tag{4.15}$$

$$Decay_h = D^{Rank_h} \times Reward_h \tag{4.16}$$

The credit of a heuristic h is given by Equation 4.17, where K is the number of heuristics. The pseudocode of this credit assignment scheme is given in Algorithm 2.

$$r_{h,t} = \frac{Decay_h}{\sum_{i=1}^{K} Decay_i} \tag{4.17}$$

Algorithm 2: Ranked Credit assignment Scheme

```
Initialize each Reward_h ← 0;
Initialize n_i ← 0;
for i = 1 to W do
    h = SlidingWindow.getIndexHer(i);
    δ_h = SlidingWindow.getImprovement(h);
    Reward_h = Reward_h + δ_h ;
end
Rank Reward_h in descending order and set Rank_h
to be the rank value of heuristic h
for h = 1 to K do
    Decay_h = D^(Rank_h) × Reward_h;
end
DecaySum = Σ_{h=1}^{K} Decay_h;
for h = 1 to K do
    r_h = Decay_h / DecaySum;
end
return r_h
```

4.7 Summary

This chapter presented a definition of the Multi-armed bandit problem, the UCB algorithm which provided the basis for the selection strategies used in this work, five MAB algorithms MAB, DMAB, SLMAB, FRRMAB and FRAMAB and four move acceptances Instantaneous, Average, Extreme and Ranked.

Three of algorithms SLMAB, FRRMAB and FRAMAB will be used as the selection strategy for the proposed hyper-heuristic. SLMAB was chosen because it introduced the concepts of the sliding window. FRRMAB was chosen because it was the latest proposed approach, to increase the adaptation it defines the quality estimation of a heuristic directly as its rewards and achieved good results. FRAMAB was the proposed selection strategy of this dissertation combining characteristics of SLMAB and FRRMAB. Regarding the credit assignment schemes, SLMAB and FRAMAB used Average and FRRMAB Ranked.

The DMAB algorithm was proposed as the selection strategy and combined with generated acceptance mechanism in [26] for the cross-domain context, and will be used for comparison.

The next chapter will present the proposed hyper-heuristic and details about its implementation.

CHAPTER 5

MAB-HH

This chapter presents details about methods used in the development and details about the hyper-heuristic implementation.

5.1 HyFlex

The HyFlex is a framework developed in Java in order to standardize the implementation of selection hyper-heuristics [2]. Its architecture is based on the dimension proposed by Cowling, Kendall and Soibeiga in [39] blocking the exchange of problem dependent information see Figure 5.1.

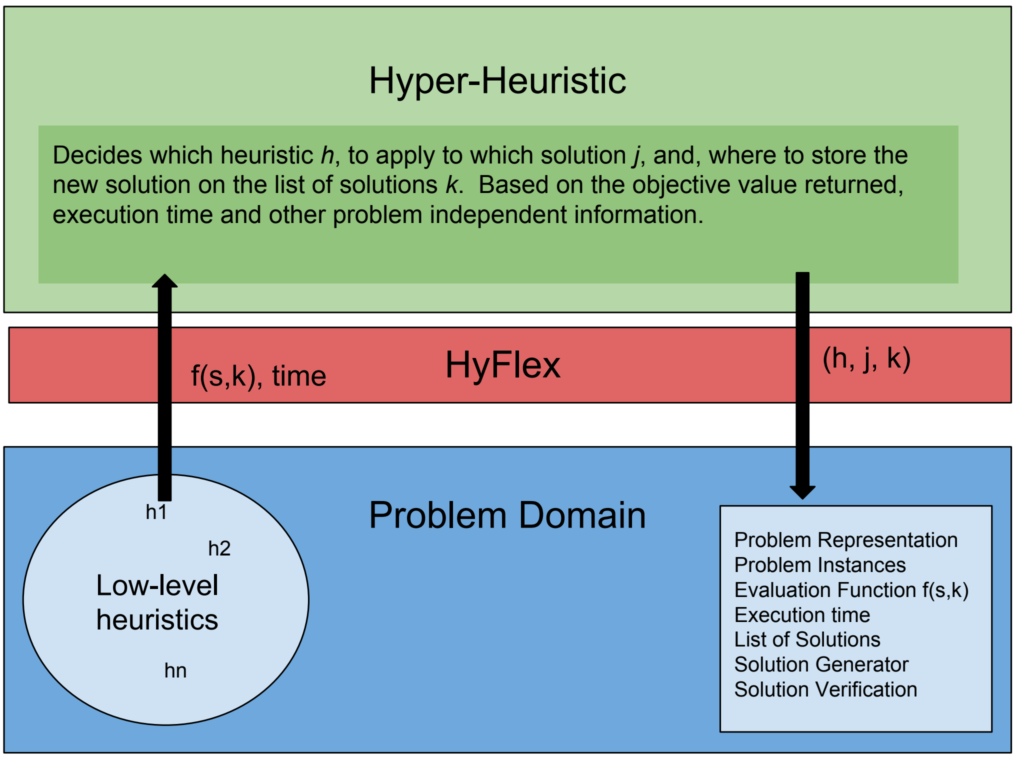

Figure 5.1: The Hyflex framework architecture with its two dimensions: Hyper-heuristics and Problem Domain. Adapted from [2].

Therefore, the Hyflex defines two classes the Hyper-heuristic and the Problem Domain.

The Hyper-heuristic class is where the hyper-heuristic is developed, and is responsible for define the stopping criteria, which is the problem domain and instance, the size of memory for the solutions (number of solutions), and most important which the selection strategy and the move acceptance mechanism will be.

The Problem Domain class defines all the domain representation. It is responsible for, defining the domain representation, the instances, the low-level heuristic, the generation of initial solutions and the solution verification.

The only informations that are passed to the Hyper-heuristic are: the value of the evaluation function, the number of low-level heuristics, its identification and the execution time of a low-level heuristic. This makes the Hyper-heuristic totally problem independent. Therefore, in order to apply the hyper-heuristic into another domain the only change necessary is on the second dimension. More details about the domains implemented and the low-level heuristics will be presented on the next section.

5.2 Problems Description

In this section the problem domains used will be presented, all of them are defined on the HyFlex framework [17]. All problems presented bellow can be formulated in several ways with different objectives and constraints. Here, they will be described as their formulation in Hyflex.

5.2.1 One Dimensional Bin Packing

The One Dimensional Bin Packing problem is a classic combinatorial optimization problem. It can be described as follows: given a set of items of a fixed weight and an infinite number of bins with fixed capacity, the objective is to pack all items using as few bins as possible [19]. The following constraints should be respected: each item can be assigned to only one bin and the total weight of the items packed in one bin should be less or equal its capacity.

The initial solution is created using the first fit heuristic, initially the items are shuffled randomly then they are packed one by one into the first bin, which they fit. And the fitness function used is presented in Equation 5.1. Where $fullnes_i$ is the sum of all the pieces in bin i, n is the number of bins and C is the capacity, this function favor bins that are filled completely, or nearly so [19]. More details about the bin packing domain can be found in [19].

$$Fitness = 1 - \left(\frac{\sum_{i=1}^{n} (fulness_i / C)^2}{n} \right) \tag{5.1}$$

On Table 5.1 the instances used and their size are presented.

Instances	***Capacity***	***Number of Items***
Instance 1	1000	2004
Instance 2	150	1000
Instance 3	100	5000
Instance 4	150	2000
Instance 5	100	5000

Table 5.1: Bin Packing Instances.

5.2.2 Boolean Satisfiability (MAXSAT)

SAT problems involves to determine if there is a configuration of boolean variables of a formula that result in the whole formula evaluating true. If such configuration exits then the formula is said to be satisfiable, otherwise, it is unsatisfiable [18]. On the MAX-SAT variation, the goal is to find the maximum number of clauses that can be satisfied on a formula by some configuration. On HyFlex the problem is formulated as a minimization, where the objective is to reduce the unsatisfied clauses.

The initial solution is created by simply randomly assigning true or false for each variable [18]. Table 5.2 presents the information about the instances used.

5.2.3 Personnel Scheduling

The Personnel Scheduling problem cannot be defined as a specific problem but as a group of problems with similar structure that differ in their constraints and objectives [20].

Instances	***Variables***	***Clauses***
Instance 1	525	2276
Instance 2	696	3122
Instance 3	525	2336
Instance 4	684	2300
Instance 5	300	1200

Table 5.2: MAXSAT Instances.

Briefly, given a set of employees with specific categories, a set of predefined tasks on a day and a set of days, the goal is to assign each employee to a specific planning period in order to meet the operational requirements [7]. The Nurse Rostering problem is implemented on HyFlex, more details about the solution generation and fitness function can be found in [20]. On Table 5.3 the information about the instances is presented.

Instances	***Staff***	***Tasks Types***	***Days***
Instance 1	25	3	30
Instance 2	54	12	42
Instance 3	51	8	42
Instance 4	21	5	31
Instance 5	16	4	31

Table 5.3: Personnel Scheduling Instances.

5.2.4 Permutation Flow Shop

The Permutation Flow Shop problem consists on assigning a set of jobs to be processed on a set of consecutive machines in a manner to minimize the completion time of the last job to exit the shop. The constraints are: each job requires a processing time on a particular machine, one machine can only process one job at time, the job ordering should be respected and a machine cannot remain idle when a job is ready to be processed [7].

The initial solution is generated by NEH algorithm [56] and the fitness function is defined by the completion time of the last job [21]. Table 5.4 presents the information of the instances for this domain.

Instances	***Number of Jobs***	***Number of Machines***
Instance 1	100	20
Instance 2	500	20
Instance 3	100	20
Instance 4	200	20
Instance 5	500	20

Table 5.4: Permutation Flow Shop Instances.

5.2.5 Traveling Salesman

The traveling salesman problem [22] is a well-known combinatorial optimization problem, given a set of cities and its distances (pairwise), where the goal is to find the shortest path where each city is visited only once and the path ends at the starting city. The objective of the optimization process is to minimize the distance traveled [7]. The initial solution is created by permutation sequences randomly generated and the fitness function is the total distance of the path. Table 5.5 presents the instances information.

Instances	***Number of Cities***
Instance 1	299
Instance 2	13509
Instance 3	575
Instance 4	2152
Instance 5	1291

Table 5.5: TSP Instances.

5.2.6 Vehicle Routing

The vehicle routing problem [23] can be defined as: given a set of customers associated with demand and service time and a set of vehicles with a fixed capacity, the goal is to design a set of routes that serve all customers respecting the following constraints. Each vehicle starts and ends at the depot, the total demand of the route does not exceed the vehicle capacity and each customer is visited once by one vehicle during its time window [7]. The fitness function is the total travel distance. Table 5.6 presents the instances information.

Instances	***Number of Vehicles***	***Vehicle Capacity***
Instance 1	250	1000
Instance 2	25	200
Instance 3	250	200
Instance 4	25	1000
Instance 5	250	200

Table 5.6: VRP Instances.

5.2.7 Low-level heuristics types and distribution

Four types of low-level heuristics are defined within HyFlex, they are:

- **Local Search** (LS): apply a perturbation in a solution over n iterations, the change is only accepted if the generated solution is better. The intensity is controlled by a parameter τ.
- **Mutation** (MU): generate a new solution by modifying the current solution by changing, removing, swapping adding or deleting one or more solution components. The mutation intensity is controlled by a parameter η.
- **Ruin-recreate** (R-R): destroy part of a solution then rebuild it using some heuristic. It is controlled by η.
- **Crossover** (CVR): combine two solutions on a new one.

On Table 5.7 the low-level heuristic distribution over the domains is presented.

Problem Domain	***LS***	***MU***	***R-R***	***CVR***	***Total***
Bin Packing	2	3	2	1	8
MAXSAT	2	4	1	2	9
TSP	6	5	1	3	15
VRP	4	4	2	2	12
Personnel Scheduling	5	1	3	3	12
Flow Shop	4	5	2	4	15

Table 5.7: Low-level heuristics distribution.

5.3 Hyper-heuristic

The proposed selection hyper-heuristic was developed using the HyFlex framework [17]. For the selection strategy, three MAB algorithms were implemented SLMAB, FRRMAB and FRAMAB. The move acceptance mechanisms used are those described on Section 1, and only one selection and acceptance are used on each run.

Another feature of the proposed selection hyper-heuristic is the on-line adaptation of low-level heuristics parameters based on reinforcement learning [9] (see Section 5.4). The proposed hyper-heuristic works on a single current solution, but also stores six auxiliary solutions including the best so far.

The pseudo-code of MAB-HH is presented in Algorithm 3. Firstly, the hyper-heuristic randomly creates an initial solution and replicates it to the auxiliary and best solutions. Afterwards, the main loop begins. Initially all low-level heuristics are run one time. Then, at each iteration, one low-level heuristic is selected using a MAB strategy and it is applied to the current solution. If the selected low-level heuristic is a crossover then one of the auxiliary solutions is randomly selected in order to perform the crossover with the current solution.

After the low-level heuristic application, its reward is calculated accordingly to the method use by the strategy and is stored in the sliding window. Then the new solution is passed to the acceptance mechanism and, if accepted, it replaces the current solution. If the new solution is also better than the best solution found so far it is set as the new best solution and randomly replaces one of the five auxiliary solutions as in [35], this process is done to create some diversity in the list. The quality and number of executions associated to the applied low-level heuristic is updated accordingly to the selection algorithm used.

When the stopping criterion is achieved (in this approach it corresponds to a time limit), the best solution found is returned as well as the number of times that each heuristic was applied, allowing the analysis of the percentage of usage of each low-level heuristic.

Algorithm 3: MAB Selection Hyper-heuristic

Input: the scaling factor C, size of window W, selection algorithm mab, acceptance A

$current_s \leftarrow createInitialSolution()$;
$best_s \leftarrow current_s$;
$crossover_{list} \leftarrow current_s$;
while *not terminated* **do**
    if *there is an unused heuristic* **then**
        $h \leftarrow uniformly\ select\ a\ not\ yet\ applied\ LLH$;
    else
        $h \leftarrow argmax_{i=1..K}(q_i + C \cdot \sqrt{\frac{2log \sum_{j=0}^{K} n_{j,t}}{n_{i,t}}})$
    end
    if *h is Crossover* **then**
        $aux_s \leftarrow randomly\ select\ one\ solution\ \in crossover_{list}$;
        $new_s \leftarrow applyHeuristic(h, current_s, aux_s)$;
    else
        $new_s \leftarrow applyHeuristic(h, current_s)$;
    end
    $r_h \leftarrow improvement(current_s, new_s)$;
    $addToWindow(r_h, h)$;
    $mab.updateHeuristicValue(h)$;
    if $moveacceptance.accept(new_s)$ **then**
        if $new_s \leq best_s$ **then**
            $best_s \leftarrow new_s$;
            $crossover_{list}.overrideRandomly(best_s)$
        end
        $current_s \leftarrow new_s$;
    end
    $parameterUpdate()$;
end
return $best_s$

5.4 Parameter adaptation of the low-level heuristics

Several types of heuristic such as mutation and crossover can compose a low-level heuristic pool. Some of these heuristics may have parameters that define its behavior (e. g. the number of perturbation made by a mutation) that have to be defined. Hence, a way of automatically set these parameters is used. This task if performed through a reinforcement learning scheme as on [9], five generic heuristic types are defined to represent the low-level heuristics, they are: $ImprovingOrEqual$, $ImprovingMore$, $WorseningMore$, $WorseningOrEqual$ and $OnlyEqual$ and the type of a low-level heuristic defines how its parameters will be updated. These types are checked from time to time in order to reflect any change, for instance a heuristic could be *ImprovingOrEqual* at the beginning of the search and change to *OnlyEqual* later on. The update rate is represented by Θ, this value is set based on the feedback obtained by the acceptance mechanism, those were found empirically on [9] and are presented on Table 5.8. The details of the parameter (par_i) update are shown in Algorithm 4.

	Feedback	Value
Θ_1	new best solution	0.01
Θ_2	better solution	0.001
Θ_3	worse solution	0.0005
Θ_4	equal solution	0.0001

Table 5.8: Θ values

Algorithm 4: Parameter Update

$u = 1$; $p \in rand(0,1)$; $par_i \in [0.2, 1]$; $\Theta = feedback$;
if $heuristic_{type} = ImprovingOrEqual$ **then**
  if $(f(S') < f(S)$ *and* $p < 0.5)$ *or* $(f(S') = f(S)$ *and* $0.25 \leq p < 0.5)$ **then**
    $u = 0$
  else if $f(S') = f(S)$ *and* $p < 0.25$ **then**
    $u = -1$
  end
else if $heuristic_{type} = ImprovingMore$ **then**
  if $(f(S') < f(S)$ *and* $0.25 \leq p < 0.5)$ *or* $(f(S') \geq f(S)$ *and* $p < 0.5)$ **then**
    $u = 0$
  else if $f(S') < f(S)$ *and* $p < 0.25$ **then**
    $u = -1$
  end
else if $heuristic_{type} = WorseningMore$ **then**
  if $f(S') < f(S)$ *and* $p < 0.5$ **then**
    $u = 0$
  else if $(f(S') < f(S)$ *and* $p < 0.5)$ *or* $(f(S') = f(S))$ **then**
    $u = -1$
  end
else
  if $f(S') = f(S)$ **then**
    $u = -1$
  end
end
return $par_i = par_i + \Theta \cdot u$

5.5 Summary

This chapter presented details about the HyFlex framework, the implementation of the hyper-heuristic, its pseudo-code and how the parameter update task was performed.

The hyper-heuristic is basically a straightforward implementation of the general framework presented on Chapter 3, this was done in order to better avail the behavior of the selection strategy and move acceptance, since great modifications could interfere on the performance. The use of stored solutions was performed to execute the crossover heuristics and the automatic parameter adaptation of the low-level heuristics was used to decrease the number of parameters to be set.

The next chapter presents the methodology of the experiments and results obtained by the approach.

CHAPTER 6

EXPERIMENTS AND RESULTS

This chapter presents the methodology used during the experimental stage of the work as well as the results obtained by the proposed approaches. In Section 6.1 the methodology used to conduct the experiments is presented. Section 6.2 presents the experiment performed in order to find a good configuration for the algorithms. The results obtained are presented in Section 6.3. Finally, Section 6.5 is presented a discussion about the algorithm performance and behavior.

6.1 Methodology

The experiments were run on an Intel Core i7 of 3.40Ghz with 6Gb of memory running the Ubuntu Linux operational system and was developed using the HyFlex framework.

In order to perform the tests the same methodology as the CHeSC 2011 Challenge [2] was used due to it be the standard methodology to mono-objective hyper-heuristics. The termination criteria of a run corresponds to 10 minutes of a 'standard' machine. [1] For each problem domain five instances were chosen and 31 runs of each algorithm were performed on each instance [2]. To compare the algorithms, a rank is created by adding the points obtained for each instance using the following score system: the top eight algorithms receive respectively 10, 8, 6, 5, 4, 3, 2 and 1 points based on the average values obtained. If there is a tie, the algorithm with the best minimum value is considered the best one; if the tie persists then the points are equally distributed among all algorithms that are tied.

[1] A benchmark application was provided by CHeSC committee in order to adjust normalized time. Therefore, the comparison between the hyper-heuristics is fair despite machine configuration differences.

6.2 MAB Configuration

This section presents details about the development and parameter configuration for the algorithms, those tests were performed to discover the impact of the parameters and to find a good configuration to be used for the comparison with the competition approaches.

The first algorithm implemented was the SLMAB and the initial goal was to test scaling factor value C in order to find one that performs well in all problem domains. The behavior of the algorithm was not the expected, since it presented great sensitivity to the parameters, for this reason, no further experiments were performed for the SLMAB, instead, the FRRMAB and FRAMAB variations were chosen. For both algorithms, besides the C value two other experiments were performed. For these experiments, two C values (a small and a high one) were selected based on the number of best averages achieved. The first experiment was related to the window size W with the goal to discover its impact regarding the quality of the solutions. The last, an experiment combining C and W values with the move acceptances presented on Chapter 3 (NA, AM, BE, OI, SA, EMC, GD, RR, ADP).

On the next sub-sections, specific details about the experiments performed for each algorithm will be presented.

6.2.1 SLMAB

Starting by the SLMAB, twelve values found empirically were used on the experiment regarding the C values, they were: (1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100). The results of the experiment are presented on Table 6.1, where the first column *Domains* is related to the problem domain, the second C shows which were the values used for the scaling value, and the other columns represents the number of the instances. The lines present the average Avg and minimum Min values in each instance for each configuration with the best values in bold.

At least, one C value has achieved one best average value on the domains, this means that there is no optimal C value regarding all instances.

		Instances									
		1		2		3		4		5	
Domains	**C**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**
BP	1	0.0743	0.0645	0.0099	0.0080	0.0061	0.0049	0.1107	0.1102	0.0087	0.0064
	5	0.0421	0.0327	0.0062	0.0036	0.0033	0.0014	0.1101	0.1096	0.0054	0.0035
	10	0.0306	0.0219	0.0037	0.0034	0.0453	0.0013	0.1103	0.1098	0.0050	0.0036
	20	0.0172	0.0137	0.0035	0.0033	0.2269	**0.0012**	0.1103	0.1095	0.0044	0.0032
	30	0.0168	0.0115	**0.0034**	0.0032	0.0487	0.0013	0.1102	0.1097	0.0046	0.0034
	40	0.0147	0.0113	0.0035	0.0032	0.1801	**0.0012**	0.1102	0.1097	0.0050	0.0032
	50	0.0133	0.0087	0.0035	0.0032	0.2504	**0.0012**	**0.1101**	**0.1093**	0.0051	0.0032
	60	0.0134	0.0088	0.0035	0.0032	0.0934	0.0013	0.1103	0.1096	**0.0042**	**0.0024**
	70	**0.0125**	**0.0083**	0.0035	0.0032	0.0032	0.0016	0.1103	**0.1093**	0.0070	0.0055
	80	0.0154	0.0087	0.0035	**0.0031**	**0.0029**	0.0014	0.1102	0.1096	0.0064	0.0046
	90	0.0154	0.0087	0.0035	0.0033	0.1416	0.0025	0.1102	0.1097	0.0065	0.0044
	100	0.0155	0.0088	0.0035	0.0033	0.1302	0.0013	0.1102	0.1096	0.0056	0.0025
MS	1	21	8	56	35	32	**2**	32	**7**	**10**	**7**
	5	25	16	53	39	35	16	36	25	12	8
	10	20	13	45	23	27	13	26	18	11	10
	20	15	7	45	18	24	9	22	14	**10**	**7**
	30	14	**4**	43	17	21	4	21	15	**10**	**7**
	40	14	7	42	19	20	7	20	11	**10**	**7**
	50	15	9	43	18	24	4	20	13	11	**7**
	60	15	7	42	15	22	7	20	13	11	9
	70	14	**4**	46	19	26	8	20	14	11	9
	80	15	5	42	14	23	7	**19**	9	11	8
	90	**13**	7	**41**	**12**	20	5	20	11	12	10
	100	14	7	**41**	**12**	**19**	4	20	14	12	8
PS	1	37	23	20593	9716	3434	3157	1988	1549	886	360
	5	27	**17**	**9763**	**9449**	3223	**3133**	1655	1470	364	315
	10	25	19	9806	9530	**3191**	3147	**1598**	**1385**	357	330
	20	26	18	9763	9501	3215	3163	1600	1450	345	325
	30	26	19	25645	19239	3231	3166	1658	1390	**345**	320
	40	27	21	26613	21453	3251	3181	1641	1441	351	**305**
	50	**26**	**17**	10413	9733	3294	3170	1655	1439	346	315
	60	30	22	10477	9564	3277	3190	1704	1478	347	325
	70	28	20	10378	9733	3261	3197	1694	1475	351	320
	80	**26**	**17**	10297	9784	3326	3166	1718	1559	346	320
	90	26	20	10355	9614	3323	3182	1680	1523	352	311
	100	29	22	10523	9637	3285	3193	1711	1575	346	315
VRP	1	118019.5	109334.9	15208.7	13529.9	262374.0	216877.3	23538.1	21947.0	203056.5	180599.2
	5	101343.5	92085.5	13471.4	12353.4	213986.4	194508.0	21491.5	20688.5	**175000.9**	**169921.3**
	10	98112.6	89114.1	13293.9	12299.9	209776.5	191194.4	21004.9	20655.0	177032.8	170669.4
	20	95354.2	87602.1	13038.6	12318.0	**208497.0**	**188987.0**	20732.8	20655.0	181620.6	173068.8
	30	92952.7	**85635.1**	12920.9	12290.2	224076.8	206332.1	20700.7	20654.4	182308.2	173225.8
	40	96947.9	89762.7	13188.2	12272.1	226026.3	204401.1	20697.8	20653.8	182535.7	174001.1
	50	98205.9	87769.1	12919.9	12311.8	225341.6	208395.7	20696.7	20652.6	184722.3	177098.6
	60	95726.3	88964.1	12981.0	12280.5	223147.4	197678.0	**20666.7**	20654.2	183185.4	176692.1
	70	**92816.9**	86492.5	12849.0	12271.2	230244.7	210752.0	20728.3	20652.5	184569.3	179675.9
	80	96348.1	89129.1	13010.4	12289.7	232519.1	207791.1	20696.9	20653.2	184149.1	176972.7
	90	96104.0	89765.0	12786.0	12280.6	234703.0	214377.9	20694.2	20654.1	183759.0	175531.9
	100	95084.9	87067.7	**12739.8**	**12265.5**	230583.1	214435.2	20693.9	**20651.1**	183879.1	177643.7
FlowShop	1	6317	6293	26912	26844	6379	6365	11486	11418	26716	26672
	5	6288	6262	26916	26854	6362	6327	11460	11415	26714	26671
	10	6282	6261	**26896**	26822	6364	6349	11472	11438	26716	26645
	20	6279	**6237**	26903	**26814**	6365	6343	**11459**	**11409**	26704	26642
	30	6280	6248	26909	26853	6362	6325	11467	11424	26713	26651
	40	6282	6249	26922	26872	**6358**	**6315**	11459	11411	26719	26679
	50	6282	6245	26911	26840	6361	6325	11466	11415	26722	26661
	60	6280	6241	26909	26860	6361	6325	11460	11426	26716	26658
	70	6281	6242	26916	26831	6361	6330	11467	11432	26715	**26630**
	80	6278	6247	26909	26856	6366	6344	11464	11423	26706	26652
	90	6281	6250	26916	26863	6364	6336	11467	11424	26716	26667
	100	**6276**	6238	26920	26868	6366	6323	11467	11413	**26702**	26654
TSP	1	49431.7	48492.8	21572115.9	21202492.1	7008.0	6897.2	71913.7	68743.1	57614.2	53904.2
	5	48475.2	48200.2	21259284.0	21080499.3	6877.4	6821.5	67872.9	**66441.2**	55192.0	53461.4
	10	48281.4	**48194.9**	**21252343.6**	**21024761.7**	6841.1	6808.9	**67575.3**	66629.9	54438.8	52644.6
	20	48267.6	**48194.9**	21300874.6	21059617.3	6831.0	6811.4	67846.8	66980.5	53874.9	52544.6
	30	48234.9	**48194.9**	21263444.4	21143013.3	6838.9	6808.8	68007.1	67367.4	**53431.1**	**52060.2**
	40	48229.2	**48194.9**	21268097.5	21249617.3	6828.9	6805.2	67967.9	66738.3	53494.1	52234.2
	50	48242.0	**48194.9**	21298582.9	21162014.2	6828.6	6804.3	68350.3	66912.1	53541.5	52418.5
	60	48240.3	**48194.9**	21259940.0	21093739.5	6830.7	6806.3	68050.8	67093.6	53520.5	52185.4
	70	**48226.7**	**48194.9**	21305597.2	21103017.8	6832.5	6798.3	68261.4	66773.8	53645.9	52075.2
	80	48227.9	**48194.9**	21305339.1	21102013.3	**6828.3**	6805.0	68253.8	67332.3	53844.2	52299.0
	90	48232.2	**48194.9**	21295899.5	21029861.3	6828.8	**6801.2**	68492.3	67559.4	53996.8	52807.6
	100	48237.0	**48194.9**	21325948.2	21129034.5	6823.9	6808.1	68291.8	67339.9	53626.7	52488.7

Table 6.1: Results of SLMAB for different scaling factors C.

With respect to the results, it is possible to affirm that the algorithm performed very well on two domains: Bin Packing and Personnel Scheduling. Using the best C of each instance, on the Bin Packing domain the algorithm achieved for $Instance$ 1 new best average and minimum, for $Instance$ 3 a new best average and for $Instance$ 5 a new best minimum. In relation to the Personnel Scheduling domain, a new best average was found on $Instance$ 3.

However, the algorithm did not present the same performance on the other four domains: MAXSAT, VRP, Flow Shop and TSP. As stated before, no further experiments were performed with this algorithm.

Finally, is was concluded that the SLMAB algorithm is very sensitive to its parameters.

6.2.2 FRRMAB

In relation to FRRMAB algorithm, the following C values were chosen (0.35, 0.5, 1, 2, 3, 4, 5, 6, 7, 8 and 9) to the experiment, the results for the scaling factor parameter are presented on Table 6.2, where the first column *Domains* is related to the problem domain, the second C shows which were the values used for the scaling value, and the other columns represents the number of the instances. The lines present the average Avg and minimum Min values in each instance for each configuration with the best values in bold.

The C values were slightly more stable being generally closer and for the MAXSAT domain a unique C values achieved the best results.

		Instances									
		1		**2**		**3**		**4**		**5**	
	C	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**
BP	0.35	0.0739	0.0500	0.0122	0.0080	0.0157	0.0090	**0.1085**	**0.1084**	0.0323	0.0145
	0.5	0.1126	0.0997	0.0092	0.0074	0.0110	0.0046	0.1134	0.1108	0.0327	0.0198
	0.7	0.0964	0.0774	0.0075	0.0069	0.0090	0.0047	0.1134	0.1113	0.0221	0.0138
	1	0.0729	0.0591	0.0062	0.0036	0.0077	0.0047	0.1120	0.1109	0.0199	0.0138
	2	0.0381	0.0251	0.0040	0.0035	0.0044	0.0028	0.1106	0.1100	0.0112	0.0065
	3	0.0304	0.0172	0.0039	0.0034	0.0041	0.0027	0.1104	0.1098	0.0101	0.0064
	4	0.0279	0.0171	0.0049	0.0035	0.0057	0.0027	0.1103	0.1097	0.0148	0.0097
	5	0.0239	0.0141	0.0038	**0.0033**	0.0039	0.0017	0.1101	0.1097	**0.0093**	0.0056
	6	0.0222	0.0113	**0.0036**	0.0035	0.0036	**0.0016**	0.1103	0.1097	0.0087	**0.0055**
	7	**0.0215**	**0.0111**	0.0041	0.0034	0.0051	0.0026	0.1104	0.1099	0.0124	0.0067
	8	0.0222	0.0137	0.0040	0.0034	**0.0035**	0.0024	0.1104	0.1098	0.0095	**0.0055**
	9	0.0230	0.0141	0.0045	0.0035	0.0038	0.0024	0.1104	0.1098	0.0094	0.0057
MS	0.35	28	**1**	58	**3**	33	**0**	26	**7**	16	**7**
	0.5	**9**	4	**23**	12	**11**	5	**13**	**7**	**9**	**7**
	0.7	10	5	31	11	13	5	15	10	10	**7**
	1	13	5	36	11	14	4	18	11	11	8
	2	14	7	39	10	21	7	20	13	13	10
	3	12	6	42	14	17	4	19	9	13	9
	4	12	4	40	11	21	6	18	11	13	9
	5	13	7	39	16	20	4	20	14	14	10
	6	14	8	41	14	15	6	20	16	13	10
	7	14	8	42	16	20	5	18	14	13	10
	8	14	6	40	16	16	7	20	13	12	9
	9	15	7	44	14	23	6	21	14	13	9
PS	0.35	27	19	10713	9477	3262	**3159**	1651	1439	374	325
	0.5	26	19	**9904**	9433	3231	**3159**	1654	**1405**	365	340
	0.7	26	19	9947	**9428**	**3212**	3165	**1620**	**1405**	357	330
	1	**25**	18	11380	9606	3268	3178	1742	1553	361	320
	2	26	**17**	10403	9558	3279	3197	2736	1480	352	**290**
	3	27	20	10326	9509	3290	3172	1773	1505	**347**	310
	4	28	21	11741	9594	3510	3221	1785	1499	361	310
	5	27	20	11870	9660	3406	3251	1784	1517	357	320
	6	26	21	10682	9546	3299	3201	1750	1470	357	320
	7	27	19	11341	9586	3535	3248	1775	1525	361	320
	8	26	19	10653	9576	3293	3171	1842	1594	357	315
	9	28	19	10408	9671	3346	3228	1807	1585	347	320
VRP	0.35	119378.5	75395.4	17256.8	13500.4	313266.6	**151624.1**	24442.3	20706.5	223060.2	**158837.8**
	0.5	106816.5	**66940.0**	13205.9	12267.0	220110.4	188044.6	20821.1	**20651.1**	181148.2	167172.4
	0.7	93801.8	72811.6	13015.7	12286.0	222388.6	209515.6	20661.9	20652.2	**175683.3**	167422.2
	1	93161.0	84058.2	12920.7	12289.8	223625.5	198241.7	20662.2	20652.5	180093.7	172557.1
	2	94343.0	87354.7	12910.8	12268.3	222506.3	204207.0	20663.7	20653.8	182536.2	176951.6
	3	94161.6	86561.8	12879.0	12273.4	**219174.5**	195896.4	**20661.1**	20651.6	180174.2	174484.9
	4	**91970.7**	81447.4	**12738.3**	12271.2	227924.0	208357.5	20693.7	20652.5	182595.7	175694.0
	5	92710.7	86619.7	12914.4	12291.2	225368.9	186354.2	20662.0	20651.8	180990.8	173364.2
	6	93377.5	82615.1	12810.9	12266.5	229296.0	205719.2	20661.8	20652.2	182974.1	177207.9
	7	93300.9	84722.8	12840.3	12280.3	238596.1	224081.7	20662.3	20651.6	185668.2	180278.4
	8	93213.6	83567.0	12879.2	12275.2	235231.2	214934.1	20661.3	20651.8	184153.5	174829.1
	9	93547.1	85997.5	12777.2	**12264.0**	237376.6	221289.2	20661.5	20651.3	184832.8	178393.0
FS	0.35	6371	6328	26958	26861	6426	6375	11552	11487	26748	26682
	0.5	6281	6254	26919	26852	6361	6323	11466	11432	26712	26674
	0.7	6281	6257	26909	26866	6362	6326	11466	11419	26706	26645
	1	6282	6249	26915	26816	6362	6327	11468	11434	26715	26653
	2	6281	6243	26908	**26808**	6358	6323	11454	11412	26705	**26609**
	3	6278	6239	**26903**	26832	6361	6323	11462	11387	26712	26654
	4	**6276**	6246	26918	26850	6365	6326	11459	**11369**	26722	26669
	5	6276	6249	**26903**	26832	6367	6323	**11450**	11406	26703	26641
	6	6277	6242	26909	26869	**6356**	**6307**	11462	11428	26706	26650
	7	6282	6253	26910	26810	6362	6323	11465	11433	26720	26666
	8	6278	**6237**	26903	26850	6359	6326	11459	11414	**26699**	26656
	9	6284	6263	26914	26850	6362	6327	11464	11432	26708	26618
TSP	0.35	49967.7	48876.8	21308028.4	21151535.6	7025.2	6971.9	69493.1	68569.6	56184.0	53981.7
	0.5	48234.4	**48194.9**	21291702.4	21090625.3	6843.6	6813.1	**68242.2**	**67211.0**	**53972.3**	**52583.4**
	0.7	48240.1	**48194.9**	**21290408.8**	**21049988.3**	**6830.4**	**6798.7**	68544.9	67791.0	54608.7	53566.9
	1	48227.1	**48194.9**	21301357.1	21086805.9	6834.3	6810.6	68828.6	68177.4	54849.9	53083.9
	2	48249.8	**48194.9**	21331459.1	21118841.2	6835.9	6814.5	69089.2	68219.1	55266.4	53490.9
	3	48237.5	**48194.9**	21362269.7	21091454.3	6835.3	6806.5	68971.7	68059.6	55382.6	53957.6
	4	48235.4	**48194.9**	21348755.2	21157482.0	6838.4	6810.3	69088.0	67759.2	55637.7	54118.9
	5	48231.5	**48194.9**	21319127.7	21122001.5	6832.7	6803.8	69131.0	67967.0	55588.7	54110.8
	6	**48219.2**	**48194.9**	21378869.9	21194705.8	6834.2	6808.8	69102.9	67904.8	55502.9	53956.0
	7	48229.7	**48194.9**	21331080.1	21131188.9	6840.0	6808.6	69334.9	68309.9	55460.7	53438.0
	8	48240.6	**48194.9**	21364813.7	21150218.7	6836.8	6801.2	69292.4	68077.7	54956.0	53669.9
	9	48238.3	**48194.9**	21358882.9	21168611.9	6834.6	6806.4	69167.8	67976.3	55370.4	53472.5

Table 6.2: Results of FRRMAB for different scaling factors C.

Regarding the results, the algorithm performed well for the Bin Packing and Personnel Scheduling, and better on the MAXSAT. For Bin Packing the algorithm achieved a new best minimum for *Instance* 1 and best average for *Instance*3. For Personnel Scheduling a new best average was achieved on *Instance* 3. The performance for the other domains VRP, Flow Shop and TSP were distant from the optimal.

With respect the window size W two values were used (100 and 500), the results are presented on Table 6.3, where the first column *Domains* is related to the problem domain, the second $C - W$ shows which were the values used for scaling factor and window size, and the other columns represents the number of the instances. The lines present the average *Avg* and minimum *Min* values in each instance for each configuration with the best values in bold.

For the FRRMAB, increasing the window size to 500, made the results worst for the majority of the scaling values. As a high value for the window size W increases the number of past rewards used on the quality estimation update of a low-level heuristic, it usually should be combined with a lower scaling factor C in order to have a better EvE balance.

		Instances									
		1		**2**		**3**		**4**		**5**	
	C-W	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**
BP	0.5-100	0.1126	0.0997	0.0092	0.0074	0.0110	0.0046	0.1134	0.1108	0.0327	0.0198
	0.5-500	0.1440	0.1277	0.0117	0.0084	0.0187	0.0103	0.1222	0.1139	0.0525	0.0262
	6-100	**0.0222**	**0.0113**	**0.0036**	**0.0035**	**0.0036**	**0.0016**	**0.1103**	**0.1097**	**0.0087**	**0.0055**
	6-500	0.0333	0.0142	0.0045	0.0036	0.0043	0.0027	0.1104	**0.1097**	0.0088	**0.0055**
MS	0.5-100	**9**	**4**	**23**	**12**	**11**	**5**	**13**	**7**	**9**	**7**
	0.5-500	19	10	48	22	30	9	21	12	13	9
	6-100	14	8	41	14	15	6	20	16	13	10
	6-500	14	5	39	14	14	**5**	18	11	12	**7**
PS	0.5-100	**26**	**19**	**9904**	**9433**	**3231**	3159	**1654**	**1405**	365	340
	0.5-500	27	20	10112	9543	3251	**3153**	1745	1502	376	**320**
	6-100	26	21	10682	9546	3299	3201	1750	1470	357	**320**
	6-500	28	20	10934	9751	3284	3195	1819	1645	**355**	335
VRP	0.5-100	106816.5	66940.0	13205.9	12267.0	**220110.4**	**188044.6**	20821.1	20651.1	**181148.2**	**167172.4**
	0.5-500	119635.5	106428.4	13033.0	12292.3	294876.8	216370.0	20958.1	20654.6	204335.6	171775.6
	6-100	**93377.5**	**82615.1**	**12810.9**	**12266.5**	229296.0	205719.2	20661.8	20652.2	182974.1	177207.9
	6-500	94930.2	86031.4	12844.5	12269.9	227681.6	205422.4	**20659.6**	**20650.8**	182913.4	176233.7
FlowShop	0.5-100	6281	6254	26919	26852	6361	6323	11466	11432	26712	26674
	0.5-500	6283	**6240**	26921	**26795**	6363	6337	11475	**11400**	26730	26667
	6-100	**6277**	6242	**26909**	26869	**6356**	**6307**	**11462**	11428	**26706**	**26650**
	6-500	6284	6261	26914	26841	6366	6327	11464	11420	26718	26661
TSP	0.5-100	48234.4	**48194.9**	**21291702.4**	21090625.3	6843.6	6813.1	68242.2	67211.0	53972.3	52583.4
	0.5-500	48234.9	**48194.9**	21317918.1	**21083724.9**	6886.5	6850.7	**68029.1**	**66920.7**	**53683.9**	**52496.0**
	6-100	**48219.2**	**48194.9**	21378869.9	21194705.8	**6834.2**	**6808.8**	69102.9	67904.8	55502.9	53956.0
	6-500	48236.1	**48194.9**	21309632.5	21114350.0	6838.4	6817.1	69290.8	67788.2	55421.2	53361.6

Table 6.3: Results of FRRMAB for different window sizes W.

The last test was performed with the move acceptance mechanism, the results are shown on Table 6.4, Table 6.5 and Table 6.6, where the first column *Domains* is related to the problem domain, the second $C - W - Accep$ shows which were the combination of scaling factor, window size and acceptance used, and the other columns represents the number of the instances. The lines present the average Avg and minimum Min values in each instance for each configuration with the best values in bold.

Using the best move acceptance for each instance, for the Bin Packing domain, the results were all very similar, the main difference could be noticed for $Instance$ 4 where the optimal value was achieved. For MAXSAT there were some improvements in the results, but they were still away from the optimal. For Personnel Scheduling, a new best average and minimum was found on $Instance$ 2 and a new best average on $Instance$ 3. For VRP new best average and minimum were found on Instances 1, 3 and 5. For the TSP and Flow Shop domains, no best value was found but some improvement could be noticed regarding the values.

The experiments showed that FRRMAB presented a similar behavior to the C value as the SLMAB. For the window size W, although the results change along with it, there was no great difference in the general performance and behavior of the algorithm. With respect to the move acceptance, it had a great influence on the performance since it is one of the cores of a selection hyper-heuristic.

		Instances									
		1		2		3		4		5	
	C-W-Accep	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**
	0.5-100-NA	0.1126	0.0997	0.0092	0.0074	0.0110	0.0046	0.1134	0.1108	0.0327	0.0198
	0.5-500-NA	0.1440	0.1277	0.0117	0.0084	0.0187	0.0103	0.1222	0.1139	0.0525	0.0262
	6-100-NA	**0.0222**	**0.0113**	0.0036	0.0035	**0.0036**	0.0016	0.1103	0.1097	**0.0087**	**0.0055**
	6-500-NA	0.0333	0.0142	0.0045	0.0036	0.0043	0.0027	0.1104	0.1097	0.0088	**0.0055**
	0.5-100-AM	0.1296	0.1149	0.0061	0.0036	0.0083	0.0026	0.1208	0.1163	0.0328	0.0119
	0.5-500-AM	0.1484	0.1327	0.0086	0.0074	0.0200	0.0083	0.1248	0.1213	0.0522	0.0189
	6-100-AM	0.0345	0.0143	0.0050	0.0035	0.0038	**0.0015**	0.1111	0.1105	0.0129	0.0076
	6-500-AM	0.0334	0.0171	0.0060	0.0036	0.0045	0.0035	0.1111	0.1104	0.0131	0.0074
	0.5-100-BE	0.0301	0.0240	0.0049	0.0033	0.0098	0.0070	0.1084	**0.1083**	0.0131	0.0097
	0.5-500-BE	0.0384	0.0288	0.0068	0.0036	0.0140	0.0090	**0.1083**	**0.1083**	0.0231	0.0126
	6-100-BE	0.0373	0.0299	0.0039	0.0033	0.0107	0.0071	0.1088	0.1087	0.0172	0.0106
	6-500-BE	0.0394	0.0300	0.0046	0.0033	0.0116	0.0080	0.1086	0.1085	0.0193	0.0118
	0.5-100-OI	0.0386	0.0295	0.0058	0.0035	0.0109	0.0080	0.1084	0.1084	0.0156	0.0117
	0.5-500-OI	0.0429	0.0317	0.0072	0.0035	0.0148	0.0093	**0.1083**	**0.1083**	0.0252	0.0149
	6-100-OI	0.0425	0.0346	0.0050	0.0033	0.0114	0.0082	0.1088	0.1086	0.0177	0.0106
	6-500-OI	0.0407	0.0320	0.0051	0.0034	0.0123	0.0081	0.1087	0.1086	0.0194	0.0147
	0.5-100-SA	0.0438	0.0293	0.0079	0.0067	0.0100	0.0061	0.1085	0.1084	0.0233	0.0107
BP	0.5-500-SA	0.0831	0.0316	0.0083	0.0067	0.0212	0.0069	0.1143	**0.1083**	0.0548	0.0242
	6-100-SA	0.0378	0.0322	0.0045	0.0033	0.0101	0.0058	0.1089	0.1086	0.0163	0.0128
	6-500-SA	0.0381	0.0271	0.0063	0.0036	0.0091	0.0058	0.1086	0.1085	0.0208	0.0118
	0.5-100-EMC	0.1294	0.1130	0.0078	0.0073	0.0088	0.0039	0.1219	0.1148	0.0336	0.0189
	0.5-500-EMC	0.1487	0.1392	0.0103	0.0080	0.0171	0.0061	0.1247	0.1206	0.0580	0.0303
	6-100-EMC	0.0312	0.0137	0.0055	0.0035	0.0045	0.0027	0.1111	0.1103	0.0122	0.0056
	6-500-EMC	0.0361	0.0196	0.0053	0.0035	0.0049	0.0028	0.1111	0.1104	0.0129	0.0086
	0.5-100-GD	0.1269	0.1151	0.0059	0.0036	0.0089	0.0028	0.1215	0.1157	0.0307	0.0171
	0.5-500-GD	0.1473	0.1367	0.0095	0.0075	0.0168	0.0050	0.1234	0.1183	0.0522	0.0181
	6-100-GD	0.0287	0.0165	0.0052	0.0035	0.0039	0.0016	0.1111	0.1105	0.0125	0.0084
	6-500-GD	0.0373	0.0252	0.0058	0.0036	0.0046	0.0028	0.1112	0.1107	0.0126	0.0066
	0.5-100-RR	0.0906	0.0684	0.0078	0.0036	0.0149	0.0095	0.1105	0.1089	0.0384	0.0170
	0.5-500-RR	0.0813	0.0682	0.0091	0.0036	0.0222	0.0135	0.1109	0.1089	0.0610	0.0400
	6-100-RR	0.0482	0.0354	0.0046	0.0035	0.0072	0.0050	0.1100	0.1093	0.0114	0.0088
	6-500-RR	0.0515	0.0430	0.0052	0.0035	0.0081	0.0058	0.1100	0.1095	0.0130	0.0097
	0.5-100-ADP	0.0560	0.0472	0.0056	0.0033	0.0122	0.0091	0.1087	0.1086	0.0228	0.0130
	0.5-500-ADP	0.0912	0.0638	0.0092	0.0068	0.0213	0.0127	0.1103	0.1087	0.0556	0.0360
	6-100-ADP	0.0366	0.0293	**0.0034**	**0.0031**	0.0102	0.0057	0.1090	0.1088	0.0170	0.0098
	6-500-ADP	0.0471	0.0405	**0.0034**	**0.0031**	0.0101	0.0050	0.1093	0.1090	0.0184	0.0130
	0.5-100-NA	9	4	23	12	11	5	13	7	**9**	**7**
	0.5-500-NA	19	10	48	22	30	9	21	12	13	9
	6-100-NA	14	8	41	14	15	6	20	16	13	10
	6-500-NA	14	5	39	14	14	5	18	11	12	**7**
	0.5-100-AM	10	5	26	9	11	5	17	10	**9**	**7**
	0.5-500-AM	18	6	49	32	33	13	23	9	13	**7**
	6-100-AM	15	7	53	32	23	4	31	13	15	11
	6-500-AM	14	6	49	12	23	7	23	17	12	8
	0.5-100-BE	10	4	22	10	14	3	**11**	6	9	7
	0.5-500-BE	10	4	31	6	16	**2**	14	8	10	8
	6-100-BE	12	7	35	9	18	3	17	12	10	8
	6-500-BE	10	4	25	9	12	4	13	8	10	8
	0.5-100-OI	12	6	44	15	20	**2**	16	9	12	9
	0.5-500-OI	15	8	47	27	29	10	19	13	13	9
	6-100-OI	16	9	49	15	27	8	22	15	15	12
	6-500-OI	16	7	43	17	25	8	20	13	14	11
	0.5-100-SA	12	3	37	13	18	4	14	7	**9**	**7**
MS	0.5-500-SA	17	4	48	36	22	**2**	20	11	**9**	**7**
	6-100-SA	**9**	3	26	6	11	4	13	9	10	**7**
	6-500-SA	10	4	**20**	**5**	**7**	**2**	12	5	10	**7**
	0.5-100-EMC	14	7	36	14	20	6	18	7	**9**	**7**
	0.5-500-EMC	20	11	48	21	28	7	22	14	13	8
	6-100-EMC	10	**2**	27	8	11	**2**	15	9	13	11
	6-500-EMC	10	4	32	9	15	3	13	**4**	11	8
	0.5-100-GD	11	4	22	7	9	4	16	10	**9**	**7**
	0.5-500-GD	22	13	50	27	32	9	22	14	14	8
	6-100-GD	16	6	52	18	23	7	36	22	13	**7**
	6-500-GD	15	6	41	11	18	5	22	12	11	9
	0.5-100-RR	22	16	51	35	37	22	24	14	13	10
	0.5-500-RR	22	15	52	18	37	19	21	15	13	9
	6-100-RR	22	14	53	19	36	17	23	17	13	10
	6-500-RR	24	13	53	31	34	13	23	17	13	9
	0.5-100-ADP	**9**	3	25	9	11	3	**11**	5	**9**	**7**
	0.5-500-ADP	12	6	39	9	21	3	15	8	**9**	**7**
	6-100-ADP	12	4	35	12	19	6	15	6	10	**7**
	6-500-ADP	9	5	30	8	12	4	13	8	**9**	**7**

Table 6.4: Results of FRRMAB for different acceptances in Domains BP-MS.

		Instances									
		1		2		3		4		5	
	C-W-Accep	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min
PS	0.5-100-NA	26	19	9904	9433	3231	3159	1654	1405	365	340
	0.5-500-NA	27	20	10112	9543	3251	3153	1745	1502	376	320
	6-100-NA	26	21	10682	9546	3299	3201	1750	1470	357	320
	6-500-NA	28	20	10934	9751	3284	3195	1819	1645	355	335
	0.5-100-AM	28	19	12150	9578	3223	3172	1670	1429	375	336
	0.5-500-AM	28	22	11442	9532	3227	3148	1726	1430	374	325
	6-100-AM	32	25	14961	9687	3472	3215	1918	1500	378	335
	6-500-AM	36	27	19156	9630	4068	3501	1902	1699	358	335
	0.5-100-BE	29	20	9614	**9205**	3323	3161	1802	1528	700	350
	0.5-500-BE	32	21	9673	9352	3370	3142	1914	1560	734	345
	6-100-BE	28	21	9559	9320	3246	3139	1828	1540	603	335
	6-500-BE	28	20	9584	9364	3311	3161	1812	1515	593	330
	0.5-100-OI	32	24	9679	9379	3310	3155	1853	1498	650	335
	0.5-500-OI	30	21	9762	9357	3317	3157	1897	1520	677	340
	6-100-OI	27	18	9558	9318	3288	3162	1745	1480	470	335
	6-500-OI	29	19	9629	9397	3294	3156	2371	1509	503	330
	0.5-100-SA	28	19	9675	9392	3245	**3136**	1861	1535	801	350
	0.5-500-SA	30	21	9648	9409	3256	3145	1838	1510	978	325
	6-100-SA	21	16	9571	9372	3219	3140	2002	1604	984	360
	6-500-SA	21	14	9559	9333	3203	3146	1812	1444	994	365
	0.5-100-EMC	25	18	9704	9501	3231	3151	1862	1403	904	325
	0.5-500-EMC	28	18	9746	9493	3276	3154	1838	1505	980	365
	6-100-EMC	**19**	**13**	9659	9402	3202	3161	1839	1497	1238	385
	6-500-EMC	20	14	9795	9560	3229	3167	1821	1450	811	345
	0.5-100-GD	25	19	9758	9535	3247	3146	1672	1440	368	340
	0.5-500-GD	26	21	9853	9423	3267	3155	1749	1445	562	340
	6-100-GD	22	16	9719	9434	3214	3161	1660	1495	347	**315**
	6-500-GD	21	14	9700	9502	**3209**	3169	**1633**	1449	**339**	**315**
	0.5-100-RR	33	23	9661	9499	3306	3152	1847	1445	961	335
	0.5-500-RR	34	19	9719	9356	3318	3169	2943	1573	1066	360
	6-100-RR	31	25	9532	9233	3304	3150	1919	1610	985	350
	6-500-RR	33	22	9637	9242	3298	3147	1891	**1400**	902	330
	0.5-100-ADP	28	19	9615	9382	3318	3160	1810	1575	649	330
	0.5-500-ADP	31	23	9659	9390	3319	3173	1899	1494	882	325
	6-100-ADP	25	15	9628	9373	3273	3140	1707	1489	412	340
	6-500-ADP	25	19	**9531**	9336	3262	3154	1799	1565	403	330
VRP	0.5-100-NA	106816.5	66940.0	13205.9	12267.0	220110.4	188044.6	20821.1	20651.1	181148.2	167172.4
	0.5-500-NA	119635.5	106428.4	13033.0	12292.3	294876.8	216370.0	20958.1	20654.6	204335.6	171775.6
	6-100-NA	93377.5	82615.1	12810.9	**12266.5**	229296.0	205719.2	20661.8	20652.2	182974.1	177207.9
	6-500-NA	94930.2	86031.4	12844.5	12269.9	227681.6	205422.4	20659.6	**20650.8**	182913.4	176233.7
	0.5-100-AM	118506.3	76752.2	13130.0	12289.2	260883.9	237189.3	20926.2	20653.8	210023.4	189502.9
	0.5-500-AM	114525.7	64115.8	13238.6	12324.2	311296.4	261093.8	20869.6	20656.5	219620.9	194175.2
	6-100-AM	98440.1	91490.5	12972.3	12267.7	245692.9	206210.9	20741.6	20653.1	190986.1	182113.6
	6-500-AM	103541.3	97389.5	12967.4	12282.3	255057.6	243240.6	20659.3	20653.2	189509.5	182590.8
	0.5-100-BE	60379.6	58730.8	13307.5	12342.1	**144083.3**	**142479.1**	20748.9	**20650.8**	146228.4	144245.3
	0.5-500-BE	61273.7	58721.5	13271.9	12302.4	145713.2	142601.1	20916.2	**20650.8**	146948.0	145145.5
	6-100-BE	60170.5	58580.9	13306.7	12348.3	145900.8	143894.8	20655.6	**20650.8**	147849.7	146423.5
	6-500-BE	60528.2	58679.7	13238.1	12272.7	146223.8	142488.6	20716.3	**20650.8**	147170.7	144572.7
	0.5-100-OI	60331.7	58740.0	13296.9	12295.2	144681.2	**142479.1**	20908.1	**20650.8**	146257.2	**144201.1**
	0.5-500-OI	61428.8	59626.4	13125.0	12301.3	145368.9	142480.1	21009.3	20652.3	146900.7	144779.9
	6-100-OI	60306.1	58418.8	13236.3	12273.1	146034.1	142600.1	20655.8	**20650.8**	148208.9	146035.5
	6-500-OI	60048.2	58550.1	12879.6	12272.5	145661.9	142484.3	**20652.2**	**20650.8**	146747.1	145060.9
	0.5-100-SA	64943.4	58998.2	13131.8	12288.2	146368.9	143915.1	20652.7	**20650.8**	**145955.5**	144623.3
	0.5-500-SA	87449.2	61309.4	12869.7	12293.3	146942.6	142546.9	20854.9	20652.5	146674.1	144614.0
	6-100-SA	**59321.7**	**57438.6**	12936.1	12274.2	145742.9	142479.2	20660.8	20651.3	147447.7	146018.8
	6-500-SA	59358.9	57767.9	13055.4	12271.2	145982.5	142525.6	20658.8	20651.8	147097.8	145709.0
	0.5-100-EMC	119595.9	82941.8	13268.5	12285.9	256992.7	241422.3	20826.2	20653.8	210664.0	188966.8
	0.5-500-EMC	119896.3	62011.8	13008.9	12295.3	315137.6	241987.9	20870.1	20654.1	215937.3	189877.9
	6-100-EMC	98640.7	91441.7	12948.6	12277.2	247347.7	224204.9	20665.6	20653.2	192091.2	181516.1
	6-500-EMC	104511.1	98200.7	13055.9	12278.1	248079.1	227878.3	20664.4	20651.8	188540.1	180620.6
	0.5-100-GD	120996.2	108114.5	13198.3	12275.1	258970.3	236691.7	20729.9	20653.3	210857.8	177113.6
	0.5-500-GD	120353.8	63661.7	**12741.9**	12291.8	291600.9	193436.7	20931.0	20654.5	209173.3	156813.2
	6-100-GD	98555.5	89720.7	12914.0	12270.9	247332.2	218493.6	20792.0	20652.2	191260.1	176304.3
	6-500-GD	101275.3	94492.6	14166.0	12281.2	248207.1	235769.5	20662.9	20652.6	189491.0	179826.6
	0.5-100-RR	62162.3	60259.5	14329.3	13357.6	148177.9	145670.5	21355.6	20654.0	148266.4	146380.5
	0.5-500-RR	62475.8	60062.9	14092.1	13311.6	149520.1	144091.6	21516.9	20654.9	149002.8	146845.9
	6-100-RR	60102.3	58700.3	14289.3	13361.9	147397.4	143837.0	21170.8	20660.0	147528.8	145066.8
	6-500-RR	60308.0	58711.4	13139.1	13353.3	147618.6	143812.6	21221.9	20657.3	147349.1	144656.4
	0.5-100-ADP	60069.5	58231.0	13355.8	12275.2	144367.0	142480.2	20683.0	**20650.8**	146160.8	144578.6
	0.5-500-ADP	61044.5	59160.1	13170.1	13299.9	145066.3	142480.2	21006.9	**20650.8**	146541.3	144960.8
	6-100-ADP	60321.4	57982.7	13063.5	12287.0	145231.3	142480.2	20654.0	20651.1	147997.3	146427.9
	6-500-ADP	60674.3	58667.0	13112.3	12282.0	145524.6	142496.6	20652.9	**20650.8**	147180.0	144915.0

Table 6.5: Results of FRRMAB for different acceptances in Domains PS-VRP.

		Instances									
		1		**2**		**3**		**4**		**5**	
	C-W-Accep	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**
	0.5-100-NA	6281	6254	26919	26852	6361	6323	11466	11432	26712	26674
	0.5-500-NA	6283	6240	26921	26795	6363	6337	11475	11400	26730	26667
	6-100-NA	6277	6242	26909	26869	6356	6307	11462	11428	26706	26650
	6-500-NA	6284	6261	26914	26841	6366	6327	11464	11420	26718	26661
	0.5-100-AM	6300	6267	26933	26862	6368	6338	11491	11454	26725	26666
	0.5-500-AM	6314	6248	26940	26876	6376	6340	11504	11401	26740	26680
	6-100-AM	6300	6272	26923	26854	6364	6323	11475	11432	26727	26650
	6-500-AM	6306	6260	26953	26861	6371	6356	11485	11438	26724	26669
	0.5-100-BE	6284	6255	26907	26845	6367	6337	11447	11393	26715	26646
	0.5-500-BE	6286	6248	26955	26879	6364	6323	11449	11399	26739	26658
	6-100-BE	6267	**6222**	26857	26816	6357	**6303**	11436	11373	26707	26607
	6-500-BE	6281	6248	26864	26796	6364	6326	**11421**	**11346**	26690	26579
	0.5-100-OI	6284	6236	26920	26859	6366	6332	11448	11410	26730	26664
	0.5-500-OI	6286	6254	26951	26874	6371	6361	11454	11405	26766	26659
	6-100-OI	6269	6223	26856	26796	6355	6306	11422	11386	26709	26654
	6-500-OI	6276	6234	26888	26786	6361	6309	11428	11372	26681	26601
	0.5-100-SA	6284	6257	26901	26849	6367	6326	11475	11419	26702	26639
FlowShop	0.5-500-SA	6298	6277	26943	26863	6371	6330	11490	11432	26728	26668
	6-100-SA	6269	6237	26869	26798	6357	6323	11446	11403	26701	26613
	6-500-SA	6271	6241	26874	26828	6357	6309	11424	11364	26687	26602
	0.5-100-EMC	6299	6273	26928	26868	6367	6323	11489	11442	26725	26666
	0.5-500-EMC	6322	6287	26937	26825	6375	6329	11513	11467	26736	26685
	6-100-EMC	6294	6254	26910	26814	6363	6326	11475	11433	26727	26664
	6-500-EMC	6303	6282	26956	26898	6371	6329	11483	11431	26727	26673
	0.5-100-GD	6273	6244	26864	26803	6357	6323	11451	11404	26678	26619
	0.5-500-GD	6277	6229	26903	26832	6364	6329	11460	11373	26713	26633
	6-100-GD	**6264**	6231	26846	26793	6359	6323	11439	11390	26669	26589
	6-500-GD	6268	6242	**26845**	**26756**	**6354**	**6303**	11427	11382	**26657**	26604
	0.5-100-RR	6343	6290	27001	26909	6386	6367	11517	11461	26790	26660
	0.5-500-RR	6350	6287	26994	26881	6383	6323	11509	11449	26809	26735
	6-100-RR	6350	6305	26980	26872	6389	6350	11499	11405	26789	26683
	6-500-RR	6341	6294	26996	26907	6391	6366	11523	11428	26788	26688
	0.5-100-ADP	6277	6247	26912	26846	6366	6328	11452	11413	26705	26614
	0.5-500-ADP	6285	6261	26943	26838	6364	6325	11468	11415	26746	26659
	6-100-ADP	6273	6236	26868	26800	6357	6323	11422	11359	26691	26623
	6-500-ADP	6273	6247	26870	26811	6360	6328	11429	11359	26680	**26584**
	0.5-100-NA	48234.4	**48194.9**	21291702.4	21090625.3	6843.6	6813.1	68242.2	67211.0	53972.3	52583.4
	0.5-500-NA	48234.9	**48194.9**	21317918.1	21083724.9	6886.5	6850.7	68029.1	66920.7	53683.9	52496.0
	6-100-NA	48219.2	**48194.9**	21378869.9	21194705.8	6834.2	6808.8	69102.9	67904.8	55502.9	53956.0
	6-500-NA	48236.1	**48194.9**	21309632.5	21114350.0	6838.4	6817.1	69290.8	67788.2	55421.2	53361.6
	0.5-100-AM	48249.1	**48194.9**	21270138.6	21040061.5	6866.0	6822.1	69088.1	67798.2	53961.2	52796.4
	0.5-500-AM	48269.2	**48194.9**	21267520.7	21153046.9	6879.8	6829.1	68018.6	66567.5	53825.2	52511.3
	6-100-AM	48272.1	**48194.9**	21482473.1	21146816.7	6866.2	6814.2	69836.4	68323.6	56389.8	54370.9
	6-500-AM	48257.6	**48194.9**	21384991.4	21197113.7	6859.8	6822.1	69674.3	67945.2	56218.8	54255.5
	0.5-100-BE	48269.8	**48194.9**	21300115.7	21097700.6	6846.7	6811.7	67790.3	67109.1	53916.6	52785.6
	0.5-500-BE	48279.9	**48194.9**	21369991.4	21114663.9	6871.8	6828.9	68056.9	66631.6	53808.1	52617.2
	6-100-BE	48226.1	**48194.9**	21247248.8	21030175.1	6830.3	6802.2	67633.0	**66414.5**	53939.3	52902.4
	6-500-BE	48244.0	**48194.9**	21271499.5	**21000851.6**	6834.3	6808.3	67640.5	67097.1	53940.5	52322.5
	0.5-100-OI	48313.9	**48194.9**	21326902.3	21081648.7	6846.0	6805.8	67866.9	67152.0	53930.9	**52232.9**
	0.5-500-OI	48252.9	**48194.9**	21395888.4	21173923.1	6854.9	6818.8	67952.2	67049.0	53984.0	52516.8
	6-100-OI	48234.9	**48194.9**	21257475.9	21051023.3	6828.7	6802.0	67490.4	66881.3	54083.0	53102.1
	6-500-OI	48231.1	**48194.9**	21283345.3	21050395.5	6838.9	6806.2	67523.1	66976.6	53879.6	52394.4
	0.5-100-SA	48230.2	**48194.9**	21290106.4	21078017.8	6836.0	6812.4	67778.9	66621.6	53659.1	52471.1
TSP	0.5-500-SA	48228.3	**48194.9**	21296835.8	21134511.2	6860.3	6820.2	67753.8	67289.1	**53445.6**	52589.8
	6-100-SA	48229.8	**48194.9**	21252402.3	21087801.5	6822.9	6804.4	67842.9	67162.5	53675.7	52805.6
	6-500-SA	48226.0	**48194.9**	21271047.3	21108516.1	6823.0	6800.2	67771.6	67118.0	53768.1	52267.9
	0.5-100-EMC	48222.3	**48194.9**	21279106.1	21126612.7	6835.9	6800.2	68105.9	67165.2	53858.6	52481.2
	0.5-500-EMC	**48213.9**	**48194.9**	21309550.6	21122345.4	6834.8	6804.1	67800.0	66553.1	54138.2	52914.8
	6-100-EMC	48225.6	**48194.9**	21249315.1	21085810.0	6825.6	**6796.9**	67934.0	66947.6	54563.1	53383.8
	6-500-EMC	48230.8	**48194.9**	21312740.1	21116151.3	6827.4	6802.3	68184.8	67438.1	54158.0	52828.3
	0.5-100-GD	48228.2	**48194.9**	21304093.8	21131757.2	6837.5	6815.9	67913.0	66987.6	53557.9	52383.2
	0.5-500-GD	48218.8	**48194.9**	21291321.2	21132722.3	6862.5	6806.9	67741.9	67127.1	53504.8	52613.4
	6-100-GD	48220.9	**48194.9**	21299621.1	21134343.9	6822.0	6802.1	68057.2	67371.2	54223.8	52976.9
	6-500-GD	48222.8	**48194.9**	**21226101.4**	21089184.8	**6819.1**	6799.4	67999.4	67231.6	54219.8	52955.7
	0.5-100-RR	50814.2	49297.5	21449943.8	21239496.7	7073.6	7011.0	70270.0	69214.1	55639.4	52950.7
	0.5-500-RR	51078.0	49259.5	21468642.5	21244681.3	7073.9	6986.9	70228.2	69223.5	55797.7	53603.2
	6-100-RR	50698.5	49384.6	21442891.9	21195494.6	7074.1	6991.0	70437.9	69190.0	55582.1	53773.9
	6-500-RR	50785.6	49482.9	21475017.7	21177211.8	7076.7	6997.5	70264.1	68923.5	55303.2	54232.9
	0.5-100-ADP	48238.9	**48194.9**	21297193.1	21109899.8	6847.0	6808.7	67742.3	66824.0	54000.0	52682.0
	0.5-500-ADP	48262.7	**48194.9**	21383125.8	21172608.0	6865.6	6821.5	68255.4	67555.4	53958.3	52660.6
	6-100-ADP	48224.1	**48194.9**	21254833.3	21041650.1	6827.5	6806.2	**67417.8**	66886.1	53749.1	52635.6
	6-500-ADP	48238.2	**48194.9**	21306732.6	21121545.7	6834.1	6804.6	67826.6	67045.3	54054.9	52975.8

Table 6.6: Results of FRRMAB for different acceptances in Domains FlowShop-TSP.

6.2.3 FRAMAB

Regarding the FRAMAB, the following C values were used (0.0001, 0.0005, 0.0008, 0,001, 0.5, 0.08 and 0.1) on the experiment, the results are presented on Table 6.7, where the first column *Domains* is related to the problem domain, the second C shows which were the values used for the scaling value, and the other columns represents the number of the instances. The lines present the average Avg and minimum Min values in each instance for each configuration with the best values in bold.

The behavior of the algorithm to the scaling factor values C was very similar to the one presented by the SLMAB, with several values achieving best results.

		Instances									
		1		2		3		4		5	
	C	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min
BP	0.1	**0.0185**	0.0142	**0.0039**	**0.0034**	0.0037	0.0026	0.1099	0.1094	**0.0057**	**0.0043**
	0.08	0.0216	0.0109	0.0040	**0.0034**	**0.0030**	**0.0016**	0.1100	0.1094	0.0063	0.0045
	0.05	0.0224	**0.0087**	0.0040	**0.0034**	0.0034	0.0024	**0.1098**	**0.1092**	0.0068	0.0045
	0.01	0.0267	0.0142	0.0047	0.0036	0.0046	0.0037	0.1103	0.1097	0.0111	0.0077
	0.008	0.0256	0.0142	0.0049	0.0035	0.0041	0.0023	0.1103	0.1095	0.0117	0.0057
	0.005	0.0245	0.0140	0.0051	0.0036	0.0044	0.0028	0.1103	0.1099	0.0110	0.0057
	0.001	0.0427	0.0325	0.0067	0.0037	0.0069	0.0057	0.1104	0.1100	0.0135	0.0106
	0.0008	0.0390	0.0278	0.0069	0.0037	0.0060	0.0046	0.1105	0.1100	0.0148	0.0098
	0.0005	0.0443	0.0306	0.0073	0.0067	0.0067	0.0049	0.1106	0.1101	0.0158	0.0106
	0.0001	0.0600	0.0488	0.0083	0.0079	0.0092	0.0062	0.1113	0.1107	0.0234	0.0171
MS	0.1	12	6	37	12	19	5	18	9	13	9
	0.08	13	4	41	15	15	6	18	12	13	10
	0.05	12	6	36	11	18	6	18	12	13	9
	0.01	12	5	39	10	15	4	16	10	12	9
	0.008	12	6	38	12	11	5	17	9	11	8
	0.005	11	6	40	13	13	5	17	11	11	**7**
	0.001	**9**	4	30	**5**	11	2	15	10	10	**7**
	0.0008	10	7	30	8	**9**	2	15	8	10	**7**
	0.0005	9	5	30	9	10	**1**	**13**	**6**	10	**7**
	0.0001	10	**3**	**26**	10	11	3	13	7	**9**	**7**
PS	0.1	24	20	9944	9645	3243	3174	1693	1440	348	310
	0.08	24	**15**	10381	**9355**	3244	3172	1780	1545	353	325
	0.05	24	18	10513	9876	3271	3186	1704	1420	351	325
	0.01	25	18	9738	9530	3231	3164	1640	1468	**351**	320
	0.008	25	18	9877	9588	3244	3172	1659	1533	353	320
	0.005	25	20	9848	9538	3260	3175	**1622**	1460	354	325
	0.001	**23**	16	9691	9478	3225	3148	1624	1438	370	325
	0.0008	24	17	9764	9535	3232	**3142**	1646	1475	361	**305**
	0.0005	24	20	**9674**	9500	3251	3159	1643	**1390**	369	310
	0.0001	26	19	9669	9400	**3224**	3156	1638	1435	379	340
VRP	0.1	96508.4	90403.9	**12715.6**	12265.2	239101.2	221913.8	20695.3	20652.6	184341.7	178016.8
	0.08	95011.3	85001.6	12875.5	12266.5	235786.5	219076.2	**20660.1**	20651.3	183290.9	174004.3
	0.05	94162.4	85754.3	12811.5	**12265.1**	229466.9	200112.0	20727.9	20651.8	182212.5	169262.1
	0.01	93523.2	85949.5	12936.5	12266.1	223321.3	205056.7	20691.6	**20651.1**	180599.7	172406.9
	0.008	93176.5	87356.8	12982.0	12279.1	224347.1	203161.1	20661.9	20652.6	181059.5	175614.5
	0.005	91760.9	81532.6	12977.2	12269.9	219874.2	201725.8	20662.0	20652.5	179078.9	171589.8
	0.001	90097.0	82796.1	12808.1	12272.0	214302.7	198654.7	20694.7	20652.5	175944.6	169384.7
	0.0008	88995.3	83761.8	13013.3	12278.6	207921.2	193305.5	20662.4	20652.2	176390.0	169055.6
	0.0005	**88000.0**	79466.1	12884.8	12266.9	204564.3	186155.8	20726.2	**20651.1**	174638.9	167807.4
	0.0001	83945.7	**78279.8**	12875.9	12284.7	**178663.1**	**163398.6**	20723.6	**20651.1**	**168000.0**	**161590.2**
FlowShop	0.1	6278	6242	26916	26867	6366	6337	11462	11416	26709	**26616**
	0.08	6277	6249	26907	26846	6363	6337	11457	11403	26711	26661
	0.05	6282	**6228**	26915	26859	6362	6326	11465	11432	26706	26664
	0.01	6286	6254	26921	26826	6367	6344	11460	11401	26718	26649
	0.008	6279	6257	26913	26831	6361	6325	11456	11423	26708	26624
	0.005	6276	6240	26915	26849	6361	6325	11465	11407	26712	26640
	0.001	6277	6244	26923	26850	6364	6326	11461	11409	26714	26672
	0.0008	6279	6251	26911	26828	6365	6330	11459	11393	26711	26664
	0.0005	6272	6240	26910	**26824**	6362	**6323**	11469	11419	26709	26639
	0.0001	**6269**	6240	**26902**	26855	**6358**	**6323**	**11455**	**11389**	**26699**	26641
TSP	0.1	48257.1	**48194.9**	21289447.6	21128059.0	6835.6	6811.0	68815.6	67805.1	54850.3	53678.0
	0.08	48238.7	**48194.9**	21311978.3	21132160.0	6836.6	6808.8	68885.6	67822.7	54823.3	53136.2
	0.05	48229.0	**48194.9**	21348611.8	21132869.3	6835.5	**6806.8**	68920.1	67801.7	55066.7	52896.0
	0.01	48244.3	**48194.9**	21296966.8	**21055057.8**	6835.2	6808.0	68537.0	67105.1	54656.5	53304.9
	0.008	**48227.5**	**48194.9**	21311848.4	21080278.6	6841.6	6817.7	68520.5	67816.7	54713.2	52943.1
	0.005	48227.9	**48194.9**	21412963.9	21158572.0	6836.2	6810.1	68348.4	67383.0	54526.0	53056.1
	0.001	48231.2	**48194.9**	21278077.1	21067675.1	6836.3	6810.4	68003.2	67105.8	54027.7	**52593.3**
	0.0008	48235.4	**48194.9**	21347083.5	21090215.9	**6827.1**	6807.4	68045.4	**66914.6**	54006.2	52755.4
	0.0005	48235.4	**48194.9**	**21271569.5**	21122071.6	6837.9	6812.1	68001.0	67217.8	54026.9	52708.6
	0.0001	48233.8	**48194.9**	21289518.0	20969515.3	6830.3	6808.8	**67658.8**	66981.1	**53753.9**	52628.2

Table 6.7: Results of FRAMAB for different scaling factors C.

With respect to the results the algorithm performed well on the Bin Packing and Personnel Scheduling domains, and achieved slightly worst results on MAXSAT. Similar to FRRMAB, for Bin Packing the algorithm achieved a new best minimum for $Instance$ 1 and best average for $Instance$3. For Personnel Scheduling a new best average was achieved on $Instance$ 3. The performance for the other domains VRP, Flow Shop and TSP were still not good.

For the window size W experiment two values were used (100 and 500), the results are presented on Table 6.8, where the first column *Domains* is related to the problem domain, the second $C-W$ shows which were the values used for scaling factor and window size, and the other columns represents the number of the instances. The lines present the average *Avg* and minimum Min values in each instance for each configuration with the best values in bold.

In contrast, to FRRMAB increasing the window size resulted in improvements for several configurations. Besides that, there was no change on the overall performance.

		Instances									
		1		**2**		**3**		**4**		**5**	
	C-W	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**
BP	0.1-100	**0.0185**	0.0142	**0.0039**	**0.0034**	0.0037	0.0026	0.1099	0.1094	**0.0057**	**0.0043**
	0.1-500	0.0197	**0.0136**	**0.0039**	**0.0034**	**0.0033**	**0.0014**	**0.1097**	**0.1092**	0.0062	0.0046
	0.0001-100	0.0632	0.0488	0.0083	0.0079	0.0092	0.0062	0.1113	0.1107	0.0234	0.0171
	0.0001-500	0.0624	0.0538	0.0084	0.0077	0.0098	0.0072	0.1107	0.1102	0.0215	0.0178
MS	0.1-100	12	6	37	12	19	5	18	9	13	9
	0.1-500	14	7	42	12	19	7	20	**1**	13	9
	0.0001-100	10	3	26	10	11	3	**13**	7	**9**	**7**
	0.0001-500	**7**	**1**	**17**	**4**	**6**	**2**	14	8	10	**7**
PS	0.1-100	**24**	20	9944	9645	3243	3174	1693	1440	**348**	**310**
	0.1-500	28	**19**	10350	9560	3303	3194	1749	1505	355	330
	0.0001-100	26	**19**	9669	**9400**	**3224**	**3156**	**1638**	1435	379	340
	0.0001-500	**24**	20	**9668**	9470	3226	**3156**	1684	1420	479	335
VRP	0.1-100	96508.4	90403.9	**12715.6**	**12265.2**	239101.2	221913.8	20695.3	20652.6	184341.7	178016.8
	0.1-500	92561.1	85665.0	12844.1	12270.1	240410.5	226426.8	**20662.7**	**20651.1**	184345.6	178580.2
	0.0001-100	**83945.7**	78279.8	12875.9	12284.7	**178663.1**	163398.6	20723.6	**20651.1**	**168000.0**	161590.2
	0.0001-500	85331.3	**74775.1**	12878.9	12278.5	187245.9	**162585.0**	20689.7	**20651.1**	170175.6	**161371.8**
FlowShop	0.1-100	6278	6242	26916	26867	6366	6337	11462	11416	26709	**26616**
	0.1-500	6277	6242	26910	26847	6359	**6318**	11463	11400	26700	26637
	0.0001-100	**6269**	**6240**	**26902**	26855	**6358**	6323	**11455**	**11389**	**26699**	26641
	0.0001-500	6278	6251	26915	**26822**	6361	6323	11457	11399	26718	26640
TSP	0.1-100	48257.1	**48194.9**	21289447.6	21128059.0	6835.6	6811.0	68815.6	67805.1	54850.3	53678.0
	0.1-500	48231.0	**48194.9**	21332664.4	21088753.4	6834.9	**6808.3**	68893.8	67761.5	55248.9	53648.2
	0.0001-100	48233.8	**48194.9**	21289518.0	**20969515.3**	**6830.3**	6808.8	**67658.8**	**66981.1**	**53753.9**	**52628.2**
	0.0001-500	**48226.4**	**48194.9**	**21286243.7**	21101238.1	6836.7	6809.0	67956.9	67203.5	54216.5	52891.5

Table 6.8: Results of FRAMAB for different window sizes W.

The results of the experiments related to the move acceptances are presented on Tables 6.9, 6.10 and 6.11, where the first column *Domains* is related to the problem domain, the second $C - W - Accep$ shows which were the combination of scaling factor, window size and acceptance used, and the other columns represent the number of the instances. The lines present the average Avg and minimum Min values in each instance for each configuration with the best values in bold.

Using the best move acceptance for each instance, on the Bin Packing the results values were very similar. An improvement can be noticed on MAXSAT for the instances 1, 2, 3 and 5. For Personnel Scheduling, a new best average and minimum was found on $Instance$ 2 and a new best average on $Instance$ 3. For the VRP a great improvement can be noticed, the algorithm achieved a new best average and minimum for $Instance$ 1, a new best average and minimum for $Instance$ 3 and a new best average and minimum for $Instance$ 5. For the rest of the domains (TSP and Flowshop), the results remained the same.

Finally, the experiments showed that FRAMAB is still sensitive to the scaling factor C parameter and have more variations regarding the window size W. Again, the move acceptance was responsible for great changes on the general behavior of the algorithm.

		Instances									
		1		**2**		**3**		**4**		**5**	
	C-W-Accep	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**
	0.1-100-NA	**0.0185**	0.0142	0.0039	0.0034	0.0037	0.0026	0.1099	0.1094	**0.0057**	**0.0043**
	0.1-500-NA	0.0197	0.0136	0.0039	0.0034	**0.0033**	**0.0014**	0.1097	0.1092	0.0062	0.0046
	0.0001-100-NA	0.0632	0.0488	0.0083	0.0079	0.0092	0.0062	0.1113	0.1107	0.0234	0.0171
	0.0001-500-NA	0.0624	0.0538	0.0084	0.0077	0.0098	0.0072	0.1107	0.1102	0.0215	0.0178
	0.1-100-AM	0.0270	0.0112	0.0041	0.0035	0.0042	0.0028	0.1109	0.1102	0.0106	0.0064
	0.1-500-AM	0.0228	**0.0107**	0.0057	0.0035	0.0042	0.0026	0.1105	0.1100	0.0103	0.0056
	0.0001-100-AM	0.0757	0.0640	0.0088	0.0077	0.0098	0.0081	0.1126	0.1119	0.0302	0.0232
	0.0001-500-AM	0.0631	0.0534	0.0087	0.0078	0.0105	0.0059	0.1119	0.1110	0.0267	0.0200
	0.1-100-BE	0.0337	0.0248	0.0036	0.0032	0.0110	0.0050	0.1089	**0.1086**	0.0161	0.0097
	0.1-500-BE	0.0351	0.0266	0.0039	0.0032	0.0102	0.0060	0.1089	**0.1086**	0.0155	0.0087
	0.0001-100-BE	0.0366	0.0271	0.0043	0.0033	0.0115	0.0081	0.1089	**0.1086**	0.0156	0.0095
	0.0001-500-BE	0.0380	0.0299	0.0038	0.0033	0.0102	0.0050	0.1090	**0.1086**	0.0163	0.0126
	0.1-100-OI	0.0340	0.0248	0.0042	0.0031	0.0112	0.0080	0.1088	0.1087	0.0161	0.0095
	0.1-500-OI	0.0347	0.0272	0.0050	0.0033	0.0119	0.0071	0.1089	0.1087	0.0164	0.0116
	0.0001-100-OI	0.0362	0.0297	0.0045	0.0034	0.0116	0.0082	**0.1088**	**0.1086**	0.0164	0.0097
	0.0001-500-OI	0.0359	0.0299	0.0042	0.0033	0.0110	0.0083	0.1089	0.1087	0.0181	0.0126
	0.1-100-SA	0.0339	0.0241	0.0042	0.0035	0.0081	0.0039	0.1088	0.1087	0.0112	0.0056
BP	0.1-500-SA	0.0363	0.0272	0.0043	0.0034	0.0067	0.0036	0.1088	0.1087	0.0117	0.0055
	0.0001-100-SA	0.0394	0.0296	0.0080	0.0069	0.0080	0.0048	0.1090	0.1087	0.0179	0.0118
	0.0001-500-SA	0.0390	0.0294	0.0081	0.0072	0.0090	0.0047	0.1090	0.1087	0.0177	0.0096
	0.1-100-EMC	0.0262	0.0136	0.0046	0.0036	0.0045	0.0024	0.1108	0.1102	0.0105	0.0075
	0.1-500-EMC	0.0211	0.0111	0.0060	0.0036	0.0038	0.0016	0.1104	0.1097	0.0098	0.0064
	0.0001-100-EMC	0.0760	0.0620	0.0084	0.0074	0.0098	0.0081	0.1125	0.1118	0.0303	0.0201
	0.0001-500-EMC	0.0657	0.0538	0.0085	0.0078	0.0098	0.0073	0.1118	0.1110	0.0268	0.0190
	0.1-100-GD	0.0266	0.0115	0.0043	0.0035	0.0043	0.0028	0.1108	0.1098	0.0099	0.0056
	0.1-500-GD	0.0235	0.0115	0.0057	0.0036	0.0037	0.0016	0.1107	0.1097	0.0087	0.0055
	0.0001-100-GD	0.0771	0.0645	0.0087	0.0078	0.0106	0.0072	0.1129	0.1115	0.0293	0.0221
	0.0001-500-GD	0.0622	0.0512	0.0084	0.0073	0.0099	0.0061	0.1120	0.1111	0.0274	0.0191
	0.1-100-RR	0.0352	0.0221	0.0044	0.0035	0.0058	0.0047	0.1101	0.1096	0.0087	0.0056
	0.1-500-RR	0.0348	0.0272	0.0041	0.0033	0.0043	**0.0014**	0.1100	0.1095	0.0067	0.0046
	0.0001-100-RR	0.0415	0.0326	0.0041	0.0035	0.0078	0.0050	0.1097	0.1092	0.0144	0.0106
	0.0001-500-RR	0.0410	0.0324	0.0051	0.0036	0.0083	0.0050	0.1099	0.1093	0.0125	0.0086
	0.1-100-ADP	0.0346	0.0242	**0.0034**	**0.0031**	0.0096	0.0060	0.1090	0.1088	0.0139	0.0088
	0.1-500-ADP	0.0361	0.0268	**0.0034**	**0.0031**	0.0090	0.0048	0.1089	0.1087	0.0130	0.0076
	0.0001-100-ADP	0.0398	0.0298	0.0036	0.0032	0.0094	0.0070	0.1091	0.1089	0.0146	0.0107
	0.0001-500-ADP	0.0377	0.0295	0.0038	0.0032	0.0095	0.0059	0.1090	0.1088	0.0148	0.0098
	0.1-100-NA	12	6	37	12	19	5	18	9	13	9
	0.1-500-NA	14	7	42	12	19	7	20	**1**	13	9
	0.0001-100-NA	10	3	26	10	11	3	13	7	9	**7**
	0.0001-500-NA	7	**1**	17	4	6	2	14	8	10	**7**
	0.1-100-AM	15	7	40	10	22	7	22	12	14	10
	0.1-500-AM	19	11	52	19	22	8	22	16	13	10
	0.0001-100-AM	11	4	24	9	11	4	12	5	9	**7**
	0.0001-500-AM	7	2	12	**3**	5	2	14	7	9	**7**
	0.1-100-BE	15	8	36	11	26	8	17	11	10	**7**
	0.1-500-BE	15	8	42	16	20	8	17	9	10	**7**
	0.0001-100-BE	8	3	25	7	14	3	10	6	10	**7**
	0.0001-500-BE	8	4	21	7	9	**1**	11	4	9	**7**
	0.1-100-OI	20	12	49	31	31	12	22	16	15	10
	0.1-500-OI	19	11	50	33	33	10	24	15	15	11
	0.0001-100-OI	11	4	41	14	19	7	19	12	12	8
	0.0001-500-OI	12	4	43	13	18	4	18	10	13	10
	0.1-100-SA	11	4	32	11	12	3	13	5	9	**7**
MS	0.1-500-SA	11	5	29	8	9	3	13	5	10	8
	0.0001-100-SA	7	2	20	7	6	2	11	4	**8**	**7**
	0.0001-500-SA	4	1	**7**	**3**	**3**	**1**	**10**	3	**8**	**7**
	0.1-100-EMC	9	5	23	8	12	2	15	4	12	9
	0.1-500-EMC	10	4	27	9	9	2	15	10	11	8
	0.0001-100-EMC	9	2	21	5	13	3	16	8	9	**7**
	0.0001-500-EMC	**5**	**1**	10	**3**	**3**	**1**	13	6	9	**7**
	0.1-100-GD	16	10	38	12	24	5	22	11	14	10
	0.1-500-GD	17	7	48	20	24	8	23	16	13	9
	0.0001-100-GD	9	5	27	8	9	4	13	8	**8**	**7**
	0.0001-500-GD	8	4	12	**3**	5	**1**	14	10	9	**7**
	0.1-100-RR	23	14	52	20	36	13	24	15	14	9
	0.1-500-RR	23	15	53	43	34	15	22	12	13	9
	0.0001-100-RR	22	12	52	26	36	11	24	17	13	9
	0.0001-500-RR	22	13	52	30	35	12	22	14	13	10
	0.1-100-ADP	12	7	36	7	17	5	16	11	10	8
	0.1-500-ADP	13	2	38	13	18	7	14	7	10	8
	0.0001-100-ADP	7	3	17	6	8	2	10	5	9	**7**
	0.0001-500-ADP	8	2	20	7	7	3	11	6	9	**7**

Table 6.9: Results of FRAMAB for different acceptances in Domains BP-MS.

		Instances									
		1		2		3		4		5	
	C-W-Accep	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**
	0.1-100-NA	24	20	9944	9645	3243	3174	1693	1440	348	**310**
	0.1-500-NA	28	19	10350	9560	3303	3194	1749	1505	355	330
	0.0001-100-NA	26	19	9669	9400	3224	3156	1638	1435	379	340
	0.0001-500-NA	24	20	9668	9470	3226	3156	1684	1420	479	335
	0.1-100-AM	28	22	12432	9940	3312	3204	1827	1464	354	325
	0.1-500-AM	30	18	13623	10016	3355	3191	1876	1630	360	325
	0.0001-100-AM	25	19	9815	9464	3219	3151	1657	1436	359	315
	0.0001-500-AM	23	15	9744	9508	**3192**	3150	1667	1459	362	320
	0.1-100-BE	27	16	9609	9405	3294	3145	1851	1578	462	335
	0.1-500-BE	27	20	9556	**9234**	3321	3158	1777	1479	515	335
	0.0001-100-BE	29	20	9630	9373	3313	3138	1789	1455	658	325
	0.0001-500-BE	33	24	9589	9311	3318	3164	1884	1500	731	370
	0.1-100-OI	28	16	9548	9270	3265	3150	1748	1420	415	325
	0.1-500-OI	29	20	9532	9251	3289	3177	1939	1580	645	345
	0.0001-100-OI	31	20	9527	9392	3295	3156	1871	1493	619	340
	0.0001-500-OI	31	19	9638	9370	3305	3155	1795	1519	695	350
	0.1-100-SA	22	13	9574	9383	3214	3157	1859	1480	1145	330
PS	0.1-500-SA	21	17	9582	9409	3209	**3136**	1927	1530	725	325
	0.0001-100-SA	26	15	9615	9392	3227	3153	1820	1473	1360	355
	0.0001-500-SA	26	20	9579	9339	3212	3141	1851	1544	1110	335
	0.1-100-EMC	21	17	9713	9570	3211	3156	1813	1468	819	330
	0.1-500-EMC	**19**	**14**	9691	9468	3214	3139	1823	1505	1070	355
	0.0001-100-EMC	26	17	9665	9371	3238	3155	1840	1475	1321	375
	0.0001-500-EMC	25	16	9634	9444	3201	3139	1912	1610	1254	370
	0.1-100-GD	21	16	9747	9506	3240	3175	**1606**	1428	**345**	315
	0.1-500-GD	22	16	9761	9471	3231	3155	2621	1485	345	325
	0.0001-100-GD	25	18	9681	9414	3239	3155	1682	1425	413	315
	0.0001-500-GD	23	17	9652	9486	3226	3152	1725	1479	466	330
	0.1-100-RR	30	20	9559	9347	3272	3161	1839	1483	809	335
	0.1-500-RR	31	18	9571	9307	3284	3164	1977	1503	1289	355
	0.0001-100-RR	31	24	**9507**	9368	3289	3148	2015	1558	773	355
	0.0001-500-RR	31	23	9583	9407	3320	3157	1958	1568	773	350
	0.1-100-ADP	26	19	9538	9352	3237	3145	1755	1520	368	325
	0.1-500-ADP	25	18	9553	9255	3266	3157	1757	**1410**	360	320
	0.0001-100-ADP	28	19	9577	9252	3253	3139	1771	1510	482	350
	0.0001-500-ADP	28	23	9565	9340	3305	3161	1769	1540	516	335
	0.1-100-NA	96508.4	90403.9	**12715.6**	12265.2	239101.2	221913.8	20695.3	20652.6	184341.7	178016.8
	0.1-500-NA	92561.1	85665.0	12844.1	12270.1	240410.5	226426.8	20662.7	20651.1	184345.6	178580.2
	0.0001-100-NA	83945.7	78279.8	12875.9	12284.7	178663.1	163398.6	20723.6	20651.1	168000.0	161590.2
	0.0001-500-NA	85331.3	74775.1	12878.9	12278.5	187245.9	162585.0	20689.7	20651.1	170175.6	161371.8
	0.1-100-AM	102888.0	91462.1	12953.0	12268.4	252700.6	231573.0	20728.4	20652.2	193246.7	187853.8
	0.1-500-AM	101666.6	97046.5	12785.5	12283.7	259311.9	242552.1	20695.5	20652.6	191709.3	186213.5
	0.0001-100-AM	101885.0	88809.0	13175.7	12285.1	208111.0	191061.2	20758.6	20651.1	176537.2	169893.1
	0.0001-500-AM	99439.6	88791.1	12987.6	12279.4	207811.2	180873.4	20824.3	20651.3	179092.3	167829.2
	0.1-100-BE	60071.7	58451.4	13179.1	12268.8	146087.9	143906.1	20657.0	**20650.8**	147606.1	145592.6
	0.1-500-BE	60267.9	58385.3	13242.6	12273.1	145697.5	**142480.2**	20658.2	**20650.8**	147659.7	145218.0
	0.0001-100-BE	60214.3	58215.1	13255.1	12313.5	145756.9	142482.1	20716.3	**20650.8**	147141.2	145303.5
	0.0001-500-BE	60360.7	58270.7	13207.1	12300.1	145631.5	142487.6	20652.2	**20650.8**	146603.1	144958.6
	0.1-100-OI	59812.4	57819.4	13221.3	12305.8	145956.3	142520.4	20688.5	**20650.8**	148175.5	145309.1
	0.1-500-OI	60192.1	58734.7	13249.6	12268.2	146454.6	142571.9	20657.0	**20650.8**	147935.6	146061.5
	0.0001-100-OI	60086.6	57892.4	13219.6	12291.4	146431.5	143933.4	20719.6	**20650.8**	147287.1	144914.1
	0.0001-500-OI	59856.9	57610.8	13275.5	12300.7	145935.1	142481.2	20651.8	**20650.8**	146873.1	144837.5
	0.1-100-SA	59786.9	57939.8	12780.8	12266.1	146099.9	143915.7	20658.9	20652.5	147051.9	144848.3
VRP	0.1-500-SA	59656.7	58503.7	12740.6	**12265.0**	146598.2	142494.2	20663.4	20652.5	147420.2	146137.7
	0.0001-100-SA	**59603.7**	57889.0	12981.0	12266.7	145855.9	143928.5	20658.6	20651.1	**146591.9**	145137.5
	0.0001-500-SA	60440.1	58410.6	12812.3	12269.6	146385.9	143902.6	20655.8	20651.1	146833.6	145015.2
	0.1-100-EMC	101959.7	93226.1	12775.8	12266.1	256789.6	230627.3	20665.3	20653.1	191906.7	185666.1
	0.1-500-EMC	102583.9	94917.8	12989.7	12278.0	260147.2	236393.2	20695.2	20653.7	192369.0	186018.4
	0.0001-100-EMC	97449.3	85791.2	12877.9	12279.7	201356.3	178562.1	20725.1	20651.1	176351.7	168600.5
	0.0001-500-EMC	94570.1	78848.6	13085.3	12287.2	209837.4	185561.3	20723.0	20652.2	176968.8	164590.8
	0.1-100-GD	102295.8	94191.3	12973.6	12286.7	257950.1	232112.9	20729.4	20653.2	192715.0	185435.3
	0.1-500-GD	102735.8	96356.4	13088.4	12279.0	261841.0	246191.5	20729.9	20651.3	192454.4	184646.8
	0.0001-100-GD	100870.4	82563.9	13179.7	12292.6	206502.2	189339.5	20724.8	20651.1	177138.8	167897.5
	0.0001-500-GD	96980.3	85609.2	13119.2	12279.8	207594.5	184578.7	20881.8	20650.8	180040.1	170748.0
	0.1-100-RR	60122.7	58495.5	14192.0	13346.5	147898.8	144076.1	21190.1	20656.5	147132.6	145955.5
	0.1-500-RR	60196.5	**57331.5**	14190.1	13334.9	146517.1	143909.3	21126.4	20657.1	147379.3	145820.7
	0.0001-100-RR	60103.9	58140.6	14133.3	13355.0	147192.5	144001.1	21253.1	20653.8	147053.7	145297.1
	0.0001-500-RR	59818.9	57877.8	14220.2	13330.6	146517.9	142499.0	21349.8	20657.3	146779.7	145259.7
	0.1-100-ADP	60107.9	59152.0	13034.4	**12265.0**	145880.1	143812.0	20655.8	**20650.8**	147999.3	145546.3
	0.1-500-ADP	60699.0	57935.6	13031.1	12267.9	**145025.6**	142480.3	20657.1	**20650.8**	147852.2	145464.0
	0.0001-100-ADP	59933.4	58065.5	13138.4	12288.3	145514.7	142512.2	**20651.1**	**20650.8**	147566.6	145836.1
	0.0001-500-ADP	60153.5	58445.2	13096.7	12280.5	146053.2	143893.6	20652.8	**20650.8**	147059.9	**144477.1**

Table 6.10: Results of FRAMAB for different acceptances in Domains PS-VRP.

		Instances									
		1		2		3		4		5	
	C-W-Accep	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**
FlowShop	0.1-100-NA	6278	6242	26916	26867	6366	6337	11462	11416	26709	26616
	0.1-500-NA	6277	6242	26910	26847	6359	6318	11463	11400	26700	26637
	0.0001-100-NA	6269	6240	26902	26855	6358	6323	11455	11389	26699	26641
	0.0001-500-NA	6278	6251	26915	26822	6361	6323	11457	11399	26718	26640
	0.1-100-AM	**6260**	6240	26928	26869	6367	6332	11483	11426	26720	26649
	0.1-500-AM	6295	6272	26930	26862	6366	6330	11480	11450	26731	26683
	0.0001-100-AM	6287	6256	26919	26843	6367	6324	11487	11444	26725	26638
	0.0001-500-AM	6295	6271	26926	26852	6370	6332	11491	11432	26725	26667
	0.1-100-BE	6274	6244	26872	26808	6362	6323	11427	11359	26694	26608
	0.1-500-BE	6277	6238	26864	26785	6358	**6311**	**11415**	11350	26688	26613
	0.0001-100-BE	6280	6251	26873	**26761**	6359	6315	11426	11375	26694	26625
	0.0001-500-BE	6274	6243	26851	26783	6364	6329	11425	11346	26679	26621
	0.1-100-OI	6274	6247	26867	26808	6354	6320	11436	11359	26701	26655
	0.1-500-OI	6278	6248	26885	26813	6357	6323	11432	11371	26701	26612
	0.0001-100-OI	6271	6247	26880	26819	6360	6323	11435	11391	26698	26643
	0.0001-500-OI	6274	6247	26869	26802	6361	6325	11433	11366	26698	26594
	0.1-100-SA	6268	6238	26869	26809	6356	6323	11437	11400	26693	26629
	0.1-500-SA	6270	6230	26870	26808	**6352**	6323	11433	11392	26693	26617
	0.0001-100-SA	6271	6238	26888	26827	6364	6325	11451	11397	26689	26600
	0.0001-500-SA	6271	**6226**	26868	26806	6358	6323	11447	11379	26691	26624
	0.1-100-EMC	6289	6258	26922	26848	6366	6323	11478	11432	26719	26659
	0.1-500-EMC	6286	6260	26936	26878	6363	6323	11479	11435	26721	26654
	0.0001-100-EMC	6282	6248	26915	26844	6363	6326	11482	11417	26725	26662
	0.0001-500-EMC	6296	6261	26921	26845	6366	6344	11480	11421	26713	26659
	0.1-100-GD	6266	6241	**26850**	26808	6354	6323	11423	11384	26662	26584
	0.1-500-GD	6268	6241	26859	26785	6355	6323	11433	11393	26666	26613
	0.0001-100-GD	6267	6239	26854	26785	6354	6323	11428	11388	**26655**	26610
	0.0001-500-GD	6270	6242	26853	26807	6356	6324	11427	11400	26658	26610
	0.1-100-RR	6343	6300	26994	26865	6390	6339	11514	11444	26785	26674
	0.1-500-RR	6348	6308	26993	26922	6390	6356	11514	11444	26757	26669
	0.0001-100-RR	6342	6307	26986	26851	6387	6361	11511	11442	26779	26680
	0.0001-500-RR	6344	6296	26991	26904	6390	6366	11525	11453	26787	26708
	0.1-100-ADP	6268	6242	26864	26780	6362	6326	11419	11383	26689	26602
	0.1-500-ADP	6269	6238	26860	26794	6361	6323	11420	11362	26696	26610
	0.0001-100-ADP	6275	6246	26867	26768	6359	6316	11425	11378	26684	**26568**
	0.0001-500-ADP	6280	6256	26868	26814	6358	6323	11424	**11338**	26684	26597
TSP	0.1-100-NA	48257.1	**48194.9**	21289447.6	21128059.0	6835.6	6811.0	68815.6	67805.1	54850.3	53678.0
	0.1-500-NA	48231.0	**48194.9**	21332664.4	21088753.4	6834.9	6808.3	68893.8	67761.5	55248.9	53648.2
	0.0001-100-NA	48233.8	**48194.9**	21289518.0	**20969515.3**	6830.3	6808.8	67658.8	66981.1	53753.9	52628.2
	0.0001-500-NA	48226.4	**48194.9**	21286243.7	21101238.1	6836.7	6809.0	67956.9	67203.5	54216.5	52891.5
	0.1-100-AM	48256.1	**48194.9**	21351538.8	21165376.0	6850.7	6804.5	69459.4	68056.3	55764.4	54299.2
	0.1-500-AM	48247.0	**48194.9**	21385591.6	21189680.2	6847.8	6822.9	69354.8	68308.8	55639.9	53962.4
	0.0001-100-AM	48235.4	**48194.9**	21302942.7	21121336.3	6840.8	6816.6	68278.7	67122.9	54538.4	52758.2
	0.0001-500-AM	48237.3	**48194.9**	21292090.4	21066761.5	6851.9	6818.3	68459.3	67733.0	54432.9	52635.8
	0.1-100-BE	48237.6	**48194.9**	21241830.0	21121798.8	6830.1	6801.6	**67297.7**	**66243.9**	53802.0	52867.4
	0.1-500-BE	48228.3	**48194.9**	21330550.4	21128346.6	6833.5	6800.1	67334.2	66847.1	53687.0	52574.0
	0.0001-100-BE	48229.8	**48194.9**	21291582.0	21071943.1	6837.1	6806.7	67554.6	66849.5	53647.6	52557.6
	0.0001-500-BE	48247.7	**48194.9**	21354330.0	21140208.4	6844.3	6816.1	67793.6	66728.9	54096.5	52842.4
	0.1-100-OI	48215.7	**48194.9**	**21220399.1**	20975971.2	6830.3	**6798.2**	67441.0	66870.2	53514.3	52760.8
	0.1-500-OI	48222.2	**48194.9**	21275340.8	21101683.7	6830.6	6807.8	67379.7	66680.4	53876.4	52445.3
	0.0001-100-OI	48242.3	**48194.9**	21311455.8	21045164.3	6841.2	6813.5	67489.6	66511.4	53915.5	52662.2
	0.0001-500-OI	48239.3	**48194.9**	21303213.6	21094528.4	6840.5	6805.6	67835.1	67162.3	54138.3	52748.6
	0.1-100-SA	48220.9	**48194.9**	21279794.5	21104663.2	6825.8	6801.9	67681.4	66671.4	53755.4	52378.4
	0.1-500-SA	48223.9	**48194.9**	21298051.2	21101517.6	**6823.4**	6800.7	67513.3	66625.7	53643.3	52574.5
	0.0001-100-SA	**48208.7**	**48194.9**	21269595.6	21085102.1	6826.4	6801.0	67743.5	67265.0	53755.0	52463.0
	0.0001-500-SA	48209.1	**48194.9**	21252985.1	21078469.9	6835.2	6810.5	67588.6	66779.2	53757.1	52723.6
	0.1-100-EMC	48227.7	**48194.9**	21328761.8	21142718.0	6825.4	6803.0	68057.0	67118.8	54226.4	52894.4
	0.1-500-EMC	48232.4	**48194.9**	21282209.5	21122791.2	6823.7	6801.5	68317.4	67460.1	54286.9	53398.7
	0.0001-100-EMC	48221.4	**48194.9**	21292161.3	21062945.1	6826.2	6804.1	67946.0	67113.0	54126.9	53087.5
	0.0001-500-EMC	48237.6	**48194.9**	21278647.4	21118576.9	6835.9	6807.6	68036.7	67392.1	54075.2	52869.0
	0.1-100-GD	48223.4	**48194.9**	21306119.6	21133760.5	6826.3	6805.9	67961.3	67263.0	53799.9	51897.4
	0.1-500-GD	48221.6	**48194.9**	21290681.9	21052104.6	6823.1	6801.6	68083.0	67284.5	53666.5	52713.7
	0.0001-100-GD	48221.0	**48194.9**	21278413.2	21126172.8	6825.6	6801.1	67928.0	66417.1	54150.3	52706.7
	0.0001-500-GD	48216.1	**48194.9**	21298949.9	21184332.7	6835.1	6816.4	68107.4	67115.1	53965.1	52780.5
	0.1-100-RR	50756.1	49259.5	21445531.1	21219799.8	7068.3	6997.5	70550.2	69370.0	55698.3	53933.1
	0.1-500-RR	50890.0	49273.7	21441988.4	21175844.5	7082.6	6981.5	70420.7	69124.0	55502.6	54064.2
	0.0001-100-RR	51037.8	49259.5	21441606.2	21065127.8	7075.9	6996.7	70446.2	69342.5	55438.2	54214.4
	0.0001-500-RR	51045.6	49297.1	21446800.2	21213839.2	7082.3	7011.8	70358.8	68879.7	55778.4	54473.8
	0.1-100-ADP	48241.9	**48194.9**	21245093.0	21017131.1	6826.7	6799.1	67545.7	66809.3	**53512.8**	**52251.9**
	0.1-500-ADP	48242.4	**48194.9**	21292131.8	21053010.4	6832.2	6803.4	67440.5	66821.8	53692.4	52588.8
	0.0001-100-ADP	48233.9	**48194.9**	21327414.4	21108840.9	6838.3	6812.5	67619.4	67025.2	53688.6	52278.1
	0.0001-500-ADP	48227.6	**48194.9**	21298953.9	21043926.9	6841.0	6803.4	67764.6	67140.0	53849.5	52694.8

Table 6.11: Results of FRAMAB for different acceptances in Domains FlowShop-TSP.

6.2.4 Comparison between FRRMAB and FRAMAB

In order to find the best configuration a comparison including all the results obtained on the previous experiments by both algorithms (FRRMAB and FRAMAB) was made, the comparison utilized the CHeSC competition score system and the results for the 10 best configurations are shown on Table 6.12, where the first column represents the *Algorithm* used, the second column presents the *Configuration* scaling factor, window size and move acceptance used and the last column *Total* represents the sum of the scores received for each configuration and algorithm in all domains.

FRAMAB achieved 8 of the best scores while FRRMAB got only 2. The best result was found by FRAMAB with $C = 0.0001$, $W = 500$ and move acceptance Simulated Annealing (SA) and the second one by FRRMAB with $C = 6$, $W = 500$ and move acceptance Great Deluge (GD).

Therefore, FRAMAB was the algorithm chosen to be compared with the other approaches to the competition.

Algorithm	**Configuration**	**Total**
FRAMAB	0.0001-500-SA	67.25
FRRMAB	6-500-GD	56.00
FRAMAB	0.1-100-GD	45.25
FRAMAB	0.1-100-NA	39.50
FRAMAB	0.0001-100-SA	39.00
FRRMAB	6-100-GD	37.50
FRAMAB	0.1-500-SA	35.00
FRAMAB	0.0001-500-EMC	33.79
FRAMAB	0.1-100-OI	33.75
FRAMAB	0.1-500-GD	31.50

Table 6.12: Comparison between the scores of MAB configurations.

6.3 Results

This section presents a comparison between FRAMAB and literature other approaches. First a comparison between the FRAMAB and the top 10 approaches from CHeSC was made, the scores obtained are presented on Table 6.13, where the first column represents the *Algorithm* used, the following columns represents the score of each domain, and the last column *Total* represents the sum of the scores received for each algorithm in all domains. The values achieved in the instances are presented on Table 6.14, where the first column *Domains* is related to the problem domain, the second *HH* represents the hyper-heuristic, and the other columns represents the number of the instances. The lines present the average *Avg* and minimum *Min* values in each instance for each configuration with the best values in bold.

FRAMAB got the following positions for the competition 5th, 4th, 11th, 9th and 1st respectively on MS, BP, PS, FlowShop, TS and VRP domains. The top three algorithms were ADAP-HH with 150, VNS-TW with 124 and ML with 105 points. FRRAMAB scored 81 points and was the 4th place algorithm.

	MS	**BP**	**PS**	**FS**	**TSP**	**VRP**	**Total**
ADAPHH	31.50	40.00	7.50	28.50	33.40	9.50	150.40
VNS-TW	31.85	5.50	34.50	35.00	13.90	3.00	123.75
ML	9.10	9.50	27.50	31.50	11.40	18.50	107.50
FRAMAB	13.15	11.00	20.00	0	3.00	34.00	81.15
PHUNTER	6.10	4.00	10.50	4.00	19.90	31.00	75.50
EPH	0	9.50	9.50	13.00	31.90	11.00	74.90
NAHH	12.20	19.50	2.00	17.00	13.90	6.00	70.60
HAHA	30.00	0	24.00	0.50	0	14.00	68.50
ISEA	1.95	30.00	13.00	1.50	12.90	1.00	60.35
KSATS	18.60	9.00	9.50	0	0	20.00	57.10
HAEA	0	7.50	2.00	4.00	9.90	19.00	42.40

Table 6.13: Scores of FRAMAB and the top ten CHeSC hyper-heuristics.

		Instances									
		1		**2**		**3**		**4**		**5**	
	HH	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**
MS	FRAMAB	7	2	20	7	6	2	11	4	8	**7**
	VNS-TW	3	1	**3**	**1**	2	1	**3**	**1**	10	**7**
	ML	5	1	10	3	3	1	9	4	8	**7**
	AdapHH	3	1	5	3	2	1	**3**	**1**	8	**7**
	KSATS	4	2	7	**1**	2	1	4	**1**	9	**7**
	EPH	7	4	11	5	6	2	14	5	13	**7**
	PHUNTER	5	1	11	5	4	2	9	4	8	**7**
	HAHA	**3**	**0**	4	**1**	**2**	**0**	5	**1**	8	**7**
	ISEA	5	2	11	4	4	1	9	4	11	8
	HAEA	6	2	12	5	5	2	12	4	11	8
	NAHH	8	5	10	5	4	2	9	5	**7**	**7**
BP	FRAMAB	0.039	0.0294	0.0081	0.0072	0.009	0.0047	0.109	0.1087	0.0177	0.0096
	VNS-TW	0.037	0.0298	0.0072	0.0036	0.0167	0.0136	0.1088	0.1087	0.0278	0.0238
	ML	0.0421	0.0323	0.0075	0.0067	0.0146	0.0124	0.1085	0.1084	0.0218	0.0178
	AdapHH	**0.0161**	**0.0131**	0.0036	**0.0028**	**0.0036**	**0.0004**	**0.1083**	**0.1083**	**0.0035**	**0.0031**
	KSATS	0.0192	0.0162	0.0078	0.0071	0.0115	0.0094	0.1089	0.1088	0.022	0.0188
	EPH	0.0504	0.043	0.0036	0.0034	0.0113	0.008	0.1087	**0.1083**	0.0224	0.0136
	PHUNTER	0.0479	0.0397	0.0036	0.0034	0.0201	0.0178	0.1091	0.1088	0.0395	0.0318
	HAHA	0.0883	0.0617	0.0073	0.0035	0.0145	0.0037	0.1102	0.1095	0.0279	0.0076
	ISEA	0.0342	0.0219	**0.0033**	0.0029	0.0037	0.0017	0.1086	0.1085	0.0064	0.0033
	HAEA	0.0452	0.0349	0.0036	0.0033	0.0138	0.0083	0.1087	0.1087	0.024	0.0168
	NAHH	0.055	0.0473	0.0035	0.003	0.0047	0.0027	0.1088	**0.1083**	0.0055	0.0046
PS	FRAMAB	26	20	**9579**	9339	**3212**	3141	1851	1544	1110	335
	VNS-TW	19	13	9628	9347	3223	**3124**	1590	1370	320	290
	ML	**18**	**11**	9812	9436	3228	3138	1605	1384	315	300
	AdapHH	24	17	9667	9435	3289	3142	1765	1448	325	295
	KSATS	22	14	9681	9405	3241	3150	1640	1410	355	315
	EPH	22	16	10074	9747	3232	3142	1615	1469	345	310
	PHUNTER	25	13	10136	9624	3255	3142	1595	**1350**	320	290
	HAHA	21	14	9666	**9325**	3236	3136	**1558**	1410	335	300
	ISEA	20	12	9966	9566	3308	3181	1660	1369	**315**	**280**
	HAEA	25	18	9795	9454	3266	3146	1699	1479	345	300
	NAHH	27	16	9827	9461	3246	3150	1644	1455	345	300
FlowShop	FRAMAB	6271	6226	26868	26806	6358	6323	11447	11379	26691	26624
	VNS-TW	6251	6230	26803	26765	6328	6303	11376	11333	**26602**	**26535**
	ML	6245	6226	**26800**	26744	6323	6304	11384	11338	26610	26559
	AdapHH	**6240**	**6214**	26814	26757	6326	6303	**11359**	**11318**	26643	26541
	KSATS	6292	6271	26860	26809	6366	6330	11466	11432	26683	26637
	EPH	6250	6232	26816	**26738**	6347	6309	11397	11328	26640	26569
	PHUNTER	6253	6221	26858	26786	6350	6303	11388	11336	26677	26600
	HAHA	6269	6246	26850	26784	6353	6323	11419	11383	26663	26603
	ISEA	6262	6241	26844	26792	6366	6308	11419	11359	26663	26590
	HAEA	6261	6244	26826	26767	6353	6318	11408	11359	26651	26600
	NAHH	6245	6222	26885	26773	**6323**	**6290**	11383	11319	26671	26590
TSP	FRAMAB	48209.1	**48194.9**	21252985.1	21078469.9	6835.2	6810.5	67588.6	66779.2	53757.1	52723.6
	VNS-TW	**48194.9**	**48194.9**	21042675.8	20848555.6	6819.1	**6796**	67378	66830.2	54028.6	52896.5
	ML	**48194.9**	**48194.9**	21093828.3	20793219.8	6820.6	6805.3	66894	66428.2	54368.4	52626.7
	AdapHH	**48194.9**	**48194.9**	**20822145.7**	**20752853.8**	**6810.5**	6797.5	66879.8	66277.1	53099.8	52383.8
	KSATS	48578.7	48365.3	21557455.9	21136630.6	6947.8	6928.3	72027.8	69214.2	58738.2	54698.1
	EPH	**48194.9**	**48194.9**	21064606.3	20941645.1	6811.9	6799.2	**66756.2**	**65958.6**	**52925.3**	**52053.4**
	PHUNTER	**48194.9**	**48194.9**	21246427.7	20754199.8	6813.6	**6796**	67136.8	66641.4	52934.4	52172
	HAHA	48414.8	48253.5	21291914.4	21060054	6918	6871.2	69324.3	68029.7	56039.9	54218.6
	ISEA	**48194.9**	**48194.9**	20868203.1	20771174.3	6832.6	6804.7	67282.1	66239.6	54129.2	52882.9
	HAEA	**48194.9**	**48194.9**	20925949.5	20766449.3	6824.6	6799.2	67488.5	66763.3	54144.9	52920.9
	NAHH	**48194.9**	**48194.9**	20971771	20747367.7	6841.8	6813.4	67418.2	66879.6	53097.7	52476.9
VRP	FRAMAB	**60440.1**	58410.6	12812.3	12269.6	146385.9	143902.6	20655.8	20651.1	**146833.6**	145015.2
	VNS-TW	76147.1	68340.4	13367.9	13298.1	148206.2	144012.6	21642.9	20651.1	149132.4	146513.6
	ML	80671.3	67622.1	13329.8	13298.4	**145333.5**	**142517**	20654.1	20651.1	148975.1	146200.8
	AdapHH	60900.6	**58052.1**	13347.6	13304.9	148516.8	145481.5	20656.6	20652.3	148689.2	146154
	KSATS	64495.5	60480.8	13296.8	12305.2	156577.9	147659.2	20655.4	20654.6	147124.6	145199.5
	EPH	74715.8	63932.2	13335.6	13284	162188.5	143510.8	**20650.8**	**20650.8**	155224.7	145976.5
	PHUNTER	64717.8	61139.3	**12290**	**12263**	146944.4	143663.9	**20650.8**	**20650.8**	148659	146472.9
	HAHA	65498.4	62794	13317.4	12298.4	155941.2	151677.6	20654.6	20651.3	148655.5	146760.6
	ISEA	70471.7	64026.7	13339.8	12310.2	149149.6	146453.8	20657.2	20651.3	150474	147544
	HAEA	60608.2	58813.8	13342.2	12313.1	146951.5	142592.6	20655.7	20652.5	147283.6	**144269.4**
	NAHH	65398.3	63475.5	13358.4	13332.6	157243	149897.9	20654.3	**20650.8**	152081.6	149898.1

Table 6.14: Results of FRAMAB and top 10 CHeSC hyper-heuristics.

The other comparison was made by adding the GEP-HH [26] to the competition pool. The scores are presented on Table 6.15 and the results are presented on Table 6.16.

There was no difference on the position achieved by the FRAMAB with the inclusion of GEP-HH, with FRRMAB still achieving the 4th place and GEP-HH the 5th. The reason for this, is that GEP-HH performed well on instances that FRRMAB did not.

Analyzing the results, FRAMAB performed very well on VRP and Personnel Scheduling domains achieving several average best values and very few minimum values. For MAXSAT and BinPacking there was no best value but the algorithm achieved good results overall; On TSP and Flow Shop domains FRAMAB got only 3 points with bad results. Although it achieved competitive results, scoring 81 points is equivalent to 27% of the total against 50% from ADAPHH. The main reason behind that is the impact of the hyper-parameters on MAB algorithms that causes the results to variate and makes it hard to find a good configuration.

	MS	**BP**	**PS**	**FS**	**TSP**	**VRP**	**Total**
ADAPHH	30.98	40.00	6.00	26.00	33.05	8.50	144.53
VNS-TW	31.85	5.00	31.00	33.00	12.55	2.00	115.40
ML	7.73	9.50	25.75	28.00	11.05	17.50	99.53
FRAMAB	12.12	11.00	20.00	0	3.00	33.00	79.12
GEP-HH	6.75	3.50	20.75	27.50	6.54	13.00	78.04
PHUNTER	5.22	4.00	8.50	1.00	18.55	30.00	67.27
EPH	0	8.50	7.50	8.00	31.54	11.00	66.54
NAHH	12.00	19.00	1.00	15.50	13.54	5.00	66.04
HAHA	29.47	0	22.00	0	0	13.00	64.47
KSATS	18.60	9.00	7.50	0	0	19.00	54.10
ISEA	1.60	29.50	11.25	0	11.54	0	53.89
HAEA	0	6.50	2.00	2.00	9.54	17.00	37.04

Table 6.15: Scores of FRAMAB, GEP-HH and the top ten CHeSC hyper-heuristics.

		Instances									
		1		**2**		**3**		**4**		**5**	
	HH	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**
MS	FRAMAB	7	2	20	7	6	2	11	4	8	**7**
	VNS-TW	3	1	**3**	**1**	2	1	**3**	**1**	10	**7**
	ML	5	1	10	3	3	1	9	4	8	**7**
	AdapHH	3	1	5	3	2	1	**3**	**1**	8	**7**
	KSATS	4	2	7	1	2	1	4	**1**	9	**7**
	EPH	7	4	11	5	6	2	14	5	13	**7**
	PHUNTER	5	1	11	5	4	2	9	4	8	**7**
	HAHA	**3**	**0**	4	**1**	**2**	**0**	5	**1**	8	**7**
	ISEA	5	2	11	4	4	1	9	4	11	8
	HAEA	6	2	12	5	5	2	12	4	11	8
	NAHH	8	5	10	5	4	2	9	5	7	**7**
	GEP-HH	5	2	13	6	3	1	9	4	**8**	**7**
BP	FRAMAB	0.039	0.0294	0.0081	0.0072	0.009	0.0047	0.109	0.1087	0.0177	0.0096
	VNS-TW	0.037	0.0298	0.0072	0.0036	0.0167	0.0136	0.1088	0.1087	0.0278	0.0238
	ML	0.0421	0.0323	0.0075	0.0067	0.0146	0.0124	0.1085	0.1084	0.0218	0.0178
	AdapHH	**0.0161**	**0.0131**	0.0036	**0.0028**	**0.0036**	**0.0004**	**0.1083**	**0.1083**	**0.0035**	**0.0031**
	KSATS	0.0192	0.0162	0.0078	0.0071	0.0115	0.0094	0.1089	0.1088	0.022	0.0188
	EPH	0.0504	0.043	0.0036	0.0034	0.0113	0.008	0.1087	**0.1083**	0.0224	0.0136
	PHUNTER	0.0479	0.0397	0.0036	0.0034	0.0201	0.0178	0.1091	0.1088	0.0395	0.0318
	HAHA	0.0883	0.0617	0.0073	0.0035	0.0145	0.0037	0.1102	0.1095	0.0279	0.0076
	ISEA	0.0342	0.0219	**0.0033**	0.0029	0.0037	0.0017	0.1086	0.1085	0.0064	0.0033
	HAEA	0.0452	0.0349	0.0036	0.0033	0.0138	0.0083	0.1087	0.1087	0.024	0.0168
	NAHH	0.055	0.0473	0.0035	0.003	0.0047	0.0027	0.1088	**0.1083**	0.0055	0.0046
	GEP-HH	0.0449	0.0346	0.0077	0.0067	0.0166	0.0135	0.1086	0.1085	0.0266	0.198
PS	FRAMAB	26	20	**9579**	9339	**3212**	3141	1851	1544	1110	335
	VNS-TW	19	13	9628	9347	3223	**3124**	1590	1370	320	290
	ML	**18**	**11**	9812	9436	3228	3138	1605	1384	315	300
	AdapHH	24	17	9667	9435	3289	3142	1765	1448	325	295
	KSATS	22	14	9681	9405	3241	3150	1640	1410	355	315
	EPH	22	16	10074	9747	3232	3142	1615	1469	345	310
	PHUNTER	25	13	10136	9624	3255	3142	1595	**1350**	320	290
	HAHA	21	14	9666	**9325**	3236	3136	**1558**	1410	335	300
	ISEA	20	12	9966	9566	3308	3181	1660	1369	**315**	**280**
	HAEA	25	18	9795	9454	3266	3146	1699	1479	345	300
	NAHH	27	16	9827	9461	3246	3150	1644	1455	345	300
	GEP-HH	19	**11**	10555	9726	3351	3142	1585	1360	315	285
Flow Shop	FRAMAB	6271	6226	26868	26806	6358	6323	11447	11379	26691	26624
	VNS-TW	6251	6230	26803	26765	6328	6303	11376	11333	**26602**	26535
	ML	6245	6226	26800	26744	6323	6304	11384	11338	26610	26559
	AdapHH	**6240**	**6214**	26814	26757	6326	6303	**11359**	**11318**	26643	26541
	KSATS	6292	6271	26860	26809	6366	6330	11466	11432	26683	26637
	EPH	6250	6232	26816	**26738**	6347	6309	11397	11328	26640	26569
	PHUNTER	6253	6221	26858	26786	6350	6303	11388	11336	26677	26600
	HAHA	6269	6246	26850	26784	6353	6323	11419	11383	26663	26603
	ISEA	6262	6241	26844	26792	6366	6308	11419	11359	26663	26590
	HAEA	6261	6244	26826	26767	6353	6318	11408	11359	26651	26600
	NAHH	6245	6222	26885	26773	**6323**	**6290**	11383	11319	26671	26590
	GEP-HH	6245	6224	**26798**	26748	6326	6303	11377	11325	26634	**26514**
TSP	FRAMAB	48209.1	**48194.9**	21252985.1	21078469.9	6835.2	6810.5	67588.6	66779.2	53757.1	52723.6
	VNS-TW	**48194.9**	**48194.9**	21042675.8	20848555.6	6819.1	**6796.0**	67378.0	66830.2	54028.6	52896.5
	ML	**48194.9**	**48194.9**	21093828.3	20793219.8	6820.6	6805.3	66894.0	66428.2	54368.4	52626.7
	AdapHH	**48194.9**	**48194.9**	**20822145.7**	**20752853.8**	**6810.5**	6797.5	66879.8	66277.1	53099.8	52383.8
	KSATS	48578.7	48365.3	21557455.9	21136630.6	6947.8	6928.3	72027.8	69214.2	58738.2	54698.1
	EPH	**48194.9**	**48194.9**	21064606.3	20941645.1	6811.9	6799.2	**66756.2**	**65958.6**	**52925.3**	**52053.4**
	PHUNTER	**48194.9**	**48194.9**	21246427.7	20754199.8	6813.6	**6796.0**	67136.8	66641.4	52934.4	52172.0
	HAHA	48414.8	**48253.5**	21291914.4	21060054.0	6918.0	6871.2	69324.3	68029.7	56039.9	54218.6
	ISEA	**48194.9**	**48194.9**	20868203.1	20771174.3	6832.6	6804.7	67282.1	66239.6	54129.2	52882.9
	HAEA	**48194.9**	**48194.9**	20925949.5	20766449.3	6824.6	6799.2	67488.5	66763.3	54144.9	52920.9
	NAHH	**48194.9**	**48194.9**	20971771.0	20747367.7	6841.8	6813.4	67418.2	66879.6	53097.7	52476.9
	GEP-HH	**48194.9**	**48194.9**	21268571.0	20845969.6	6826.6	6805.8	67105.2	66549.8	54755.3	53174.4
VRP	FRAMAB	**60440.1**	58410.6	12812.3	12269.6	146385.9	143902.6	20655.8	20651.1	**146833.6**	145015.2
	VNS-TW	76147.1	68340.4	13367.9	13298.1	148206.2	144012.6	21642.9	20651.1	149132.4	146513.6
	ML	80671.3	67622.1	13329.8	13298.4	**145333.5**	**142517.0**	20654.1	20651.1	148975.1	146200.8
	AdapHH	60900.6	**58052.1**	13347.6	13304.9	148516.8	145481.5	20656.6	20652.3	148689.2	146154.0
	KSATS	64495.5	60480.8	13296.8	12305.2	156577.9	147659.2	20655.4	20654.6	147124.6	145199.5
	EPH	74715.8	63932.2	13335.6	13284.0	162188.5	143510.8	**20650.8**	**20650.8**	155224.7	145976.5
	PHUNTER	64717.8	61139.3	**12290.0**	**12263.0**	146944.4	143663.9	**20650.8**	**20650.8**	148659.0	146472.9
	HAHA	65498.4	62794.0	13317.4	12298.4	155941.2	151677.6	20654.6	20651.3	148655.5	146760.6
	ISEA	70471.7	64026.7	13339.8	12310.2	149149.6	146453.8	20657.2	20651.3	150474.0	147544.0
	HAEA	60608.2	58813.8	13342.2	12313.1	146951.5	142592.6	20655.7	20652.5	147283.6	**144269.4**
	NAHH	65398.3	63475.5	13358.4	13332.6	157243.0	149897.9	20654.3	**20650.8**	152081.6	149898.1
	GEP-HH	83294.9	69895.6	13337.9	13312.9	145418.9	145481.5	20653.8	20651.9	149007.9	146471.4

Table 6.16: Results of FRAMAB, GEP-HH and top 10 CHeSC hyper-heuristics.

6.4 Low-Level Heuristics Selection

By analyzing the low-level heuristic selection of a hyper-heuristic we can better understand its behavior and the performance of the low-level heuristics. Two breakpoints were used to gather information about the low-level heuristic selection on the beginning and on the middle, this was done by dividing the total execution time (in this case, 10 minutes) by 3 (first breakpoint) and 2 (second breakpoint), the percentages showed below are the average of choices for all instances. For the figures, the numbers on the labels are related to the id of the low-level heuristic in HyFlex, and in the slices show its selection's percentage regarding all instances of the domain.

Starting by MAXSAT domain the percentages are presented on Figure 6.1, the most selected heuristic was h_{10} with 16.62%, followed by h_9 with 11.71% and h_1 with 11.08%, in relation to the breakpoints no big changes were noticed. The two first ones h_{10} and h_9 are crossovers and h_1 is a mutation.

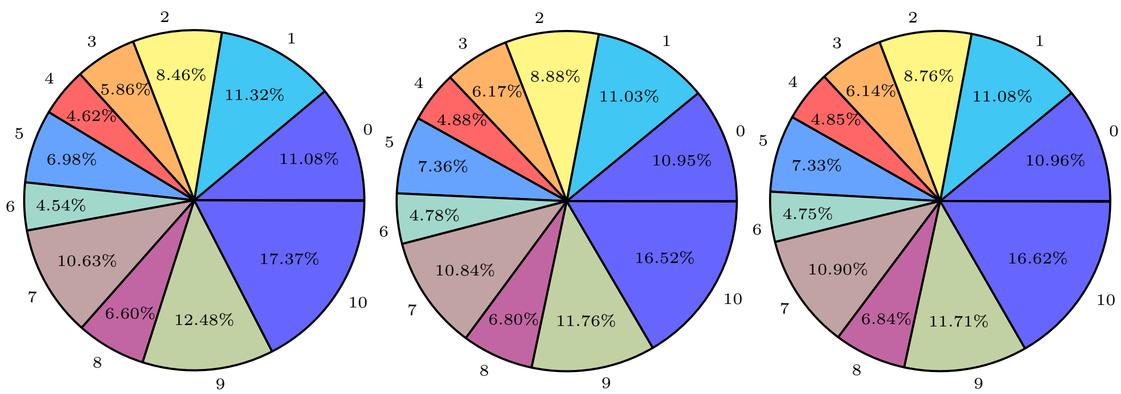

Figure 6.1: Percentage of Selection of the three breakpoints for MAXSAT.

Regarding Bin Packing its percentages are shown on Figure 6.2. The most selected heuristics was h_2 with 19.42%, h_7 with 17.15% and h_1 with 14.18%. h_7 is a crossover, h_1 and h_2 are ruin-recreate heuristics.

For Personnel Scheduling the percentages are presented on Figure 6.3. The most selected heuristic was h_4 with 22.60% followed by h_3 with 18.49% and h_2 with 15.80%. All of them are Local Search ones.

With respect to Flow Shop its percentages are presented on Figure 6.4. The most

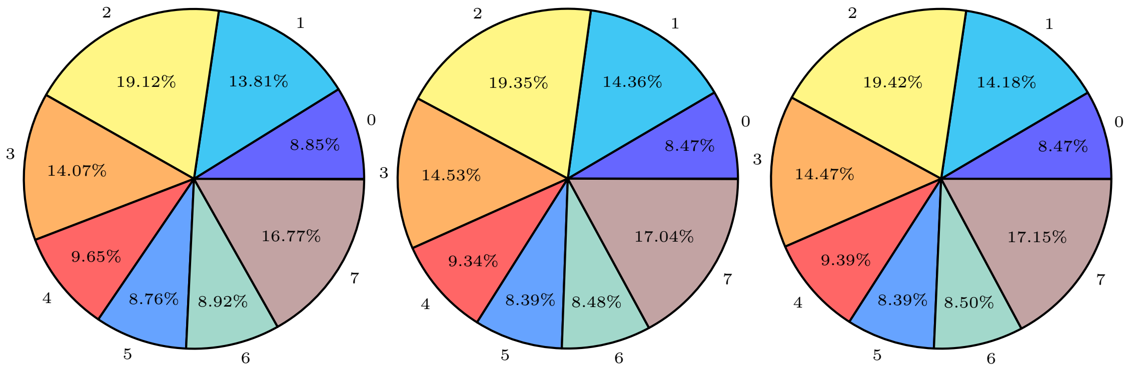

Figure 6.2: Percentage of Selection of the three breakpoints for Bin Packing.

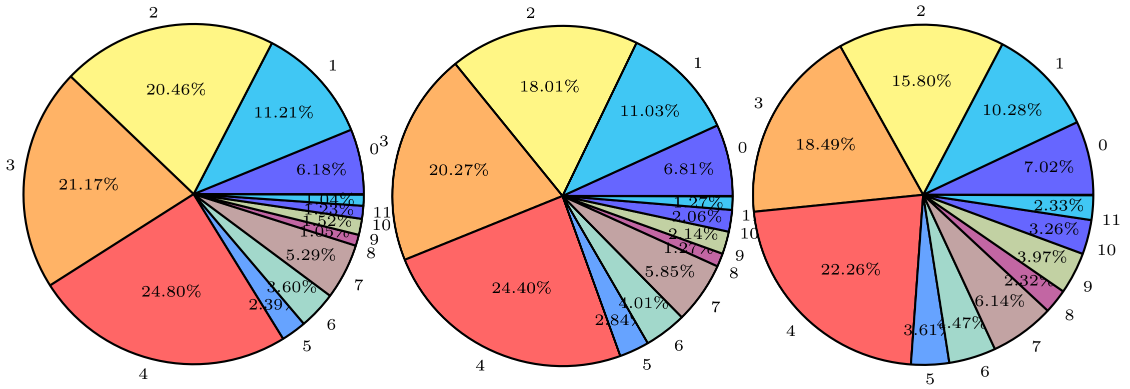

Figure 6.3: Percentage of Selection of the three breakpoints for Personnel Scheduling.

selected heuristic was h_{11} with 9% followed by h_{13} with 8.34% and h_7 with 7.20%. Talking about the heuristic types, h_9 is a local search and h_{11} and h_{13} are crossovers. An interesting thing about this domain is that all heuristics have almost the same selection percentage.

For TSP the percentages are showed on Figure 6.5. The most selected heuristic was h_{10} with 15.66% followed by h_9 with 13.54% and h_{12} with 11.11%. Regarding the heuristic types, h_{10} and h_{12} are crossovers and h_9 is a local search.

Finally, on Figure 6.6 the percentages of VRP are presented. The most selected heuristic was h_5 with 20.03%, followed by h_3 with 12.88% and h_6 with 11.90%. h_5 and h_6 are crossovers and h_3 a ruin-recreate.

Overall, the selection percentages did not change much over the time. This means that the low-level heuristics chosen at the beginning are those that continue until the end of the search. A possible reason for this behavior is that as the search progresses it gets more difficult to find new improving solutions. For that reason, the majority of the

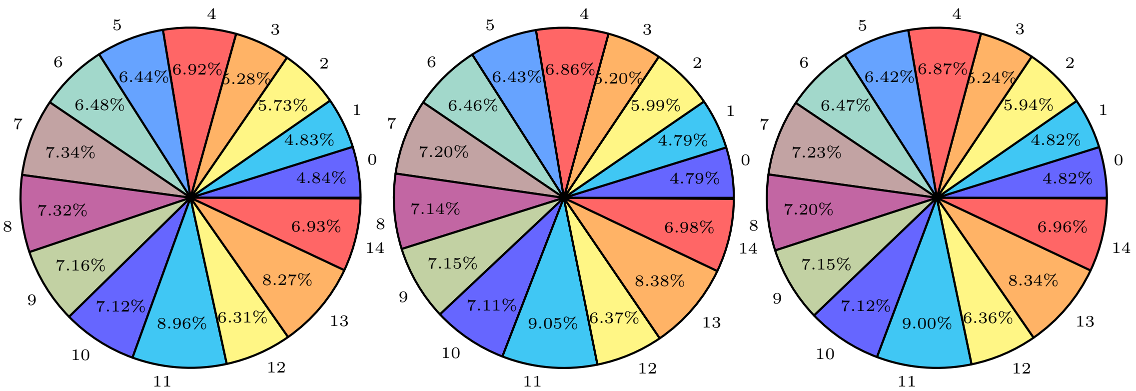

Figure 6.4: Percentage of Selection of the three breakpoints for Flow Shop.

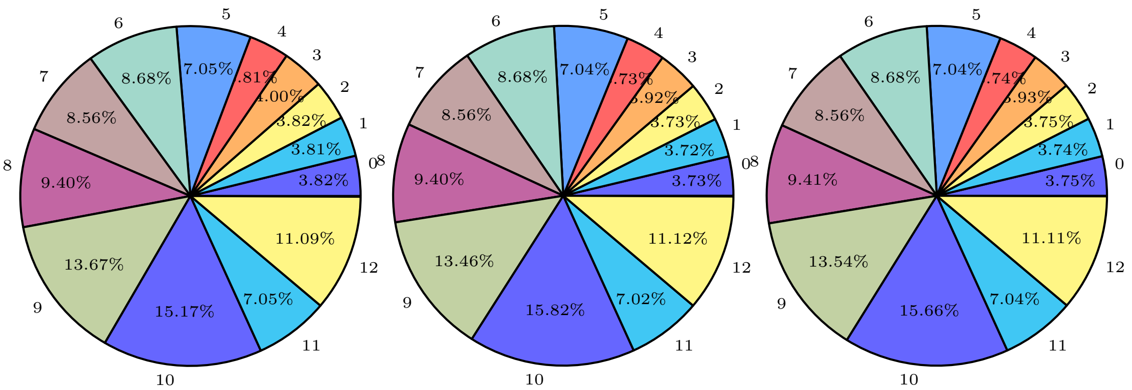

Figure 6.5: Percentage of Selection of the three breakpoints for TSP.

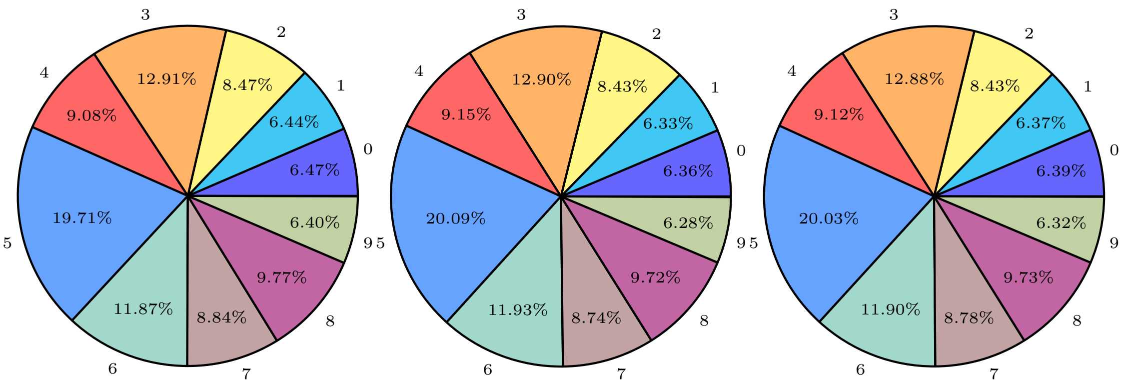

Figure 6.6: Percentage of Selection of the three breakpoints for VRP.

rewards on the window will be equal to 0, so the selection is done almost fully based on the exploration term of the algorithm.

Another characteristic noticed is the preference for crossover heuristics and ruin-recreate heuristics to the majority of domains. This shows that those types of heuristic usually perform well throughout the search, but may not achieve the best results on its own. Perhaps combining those heuristics with mutation ones could improve the results.

6.5 Summary

In this chapter, the experiments and results obtained by the proposed hyper-heuristic were presented.

Starting with the parameters setting experiments, is interesting to note that the performance of all algorithms is very similar in general, since all of them performed well on some domains/instances but have great fluctuations on their results. The reason behind that is the impact that all parameters tested (scaling factor, window size and move acceptance) had over the algorithms. In fact, there was almost a different optimal value for each parameter in each instance, which makes hard to configure a good setup and affects the application of the MAB selection method over different domains as seen on the comparison with CHeSC approaches. Over the setup of the parameters several new best values were obtained, but when the final configuration was chosen some of them were lost (since the configurations that achieved it were not used).

The results obtained by the FRAMAB were competitive but due to the sensitivity to the parameters discussed above, its behavior was not the expected since there was great variance on its results over the instances. Regarding the low-level heuristic selection percentage, it seems that the algorithm decides what are the most promising heuristic at the beginning of the search progress and those low-heuristics are applied until the end. This shows that the algorithm does not change its behavior according to the state of the search. Another point is the preference of crossover and ruin-recreate low-level heuristics over the rest. Those two points can be explained by the earlier good results of these types of low-level heuristics and the low scaling value used which increases the exploitation behavior of the algorithm.

Overall, MAB algorithms performed well but it is possible to note that the parameters are domain dependent, hence, to achieve its better performance they need to be set for each instance. The main issue is that the exploitation vs exploration behavior is defined mainly by the scaling factor value and not by the characteristics of the instances, this causes an erratic behavior of the algorithm: good and bad results even for instances in the same domain.

During the duration of this empirical research several experiments (such as other window sizes, changes in the policy of the sliding windows, different acceptances, etc) were performed in order to find which values and parameters use, but they were not presented here. This was done in order to present a clearer idea of the goal of the dissertation. Not all configurations could be experimented. The reason of that was the time necessary for conducting the experiments. Given 1 algorithm, 1 scaling factor value, 1 window size value and 9 acceptances, since each run lasts 8 minutes (time defined by the benchmark tool), and 31 runs are made for the 6 domains with 5 instances each one, the time to perform this unique test in a sequential way is of 66960 minutes or 46.5 days. In this work, 3 algorithms were used. Thus, only the experiments which lead to promising results were continued but even with those, not all variations could be experimented.

CHAPTER 7

CONCLUSION

Hyper-heuristics are a recent research field that aims to be both: effective and adaptive. In order to achieve these goals, the design of a hyper-heuristic can be very difficult leading to complex approaches.

This dissertation focused on selection hyper-heuristics (which is the most common type), that are hyper-heuristics that selects at each step a low-level heuristic to apply according some strategy. The strategy used here was based on MAB algorithms that are usually applied in a similar context, the Adaptive Operator Selection.

Three MAB algorithms were used they were: SLMAB, FRRMAB and FRAMAB. The first two are well-known algorithms with several works, the last one is a slight variation of FRRMAB proposed here. In order to analyze the performance and behavior of the algorithms an empirical experiment using the CHeSC methodology was performed, and the results compared with top ten best performant hyper-heuristics from CHeSC.

First an experiment was made exploring different values for the parameters of the algorithms in order to find a good configuration. The parameter varied were the scaling factor, window size. Then, a combination of several move acceptance mechanisms was also performed. These experiments showed that all strategies had a similar behavior, having basically the same set of good and bad traits. The selection strategies were highly affected by the parameters and presented great fluctuations on their result values. The CheSC score system was used in order to find the best configuration. The FRAMAB strategy with Simulated Annealing acceptance was the winner.

The proposed hyper-heuristic achieved very good results in some domains proving to be effective but did not achieve the same performance on several others lacking adaptability. The main reason for that was the high influence of the parameters. Another point was how the move acceptance guides the selection mechanism through the search space creating great changes on the performance of the algorithm. Those points show how complex the development of a hyper-heuristic can be, especially with respect to generality and efficiency across several domains. Its valid to note that even ADAPHH one of the best performing selection hyper-heuristics for cross-domain performed well only in half of the instances, this shows how hard the cross-domain applications can be.

Finally, it is concluded that a selection strategy based on Multi-Armed Bandit concepts can be successful, but several points remain for further exploration in order to decrease the sensitive of MAB algorithms to their parameters.

Future works include the investigation of ways to better adapt the parameters, the use of other selection architectures and new move acceptances.

BIBLIOGRAPHY

[1] E. K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and J. R. Woodward, "A classification of hyper-heuristic approaches," in *Handbook of metaheuristics*. Springer, 2010, pp. 449–468.

[2] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, B. McCollum, G. Ochoa, A. J. Parkes, and S. Petrovic, "The cross-domain heuristic search challenge–an international research competition," in *Learning and Intelligent Optimization*. Springer, 2011, pp. 631–634.

[3] A. S. Ferreira, A. T. Pozo, and R. A. Gonçalves, "Aplicação do algoritmo aco-hh para o problema de cobertura de conjuntos," 2010.

[4] A. S. Ferreira, P. Aurora, and R. A. Gonçalves, "An ant colony based hyper-heuristic approach for the set covering problem," *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 4, no. 1, pp. 1–21, 2015.

[5] E.-G. Talbi, *Metaheuristics: from design to implementation*. John Wiley & Sons, 2009, vol. 74.

[6] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Tradeoffs in the empirical evaluation of competing algorithm designs," *Annals of Mathematics and Artificial Intelligence*, vol. 60, no. 1-2, pp. 65–89, 2010.

[7] N. Sabar, M. Ayob, G. Kendall, and R. Qu, "The automatic design of hyper-heuristic framework with gene expression programming for combinatorial optimization problems," 2014.

[8] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Ozcan, and R. Qu, "Hyper-heuristics," *J Oper Res Soc*, vol. 64, no. 12, pp. 1695–1724, Dec 2013. [Online]. Available: http://dx.doi.org/10.1057/jors.2013.71

[9] M. Mısır, K. Verbeeck, P. De Causmaecker, and G. V. Berghe, “An intelligent hyper-heuristic framework for chesc 2011,” in *Learning and Intelligent OptimizatioN.* Springer, 2012, pp. 461–466.

[10] H.-L. F. P. Ross and D. Corne, “A promising hybrid ga/heuristic approach for open-shop scheduling problems,” in *Proc. 11¡ sup¿ th¡/sup¿ European Conference on Artificial Intelligence*, 1994, pp. 590–594.

[11] E. Burke, G. Kendall, D. L. Silva, R. O’Brien, and E. Soubeiga, “An ant algorithm hyperheuristic for the project presentation scheduling problem,” in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 3. IEEE, 2005, pp. 2263–2270.

[12] E. K. Burke, B. McCollum, A. Meisels, S. Petrovic, and R. Qu, “A graph-based hyper-heuristic for educational timetabling problems,” *European Journal of Operational Research*, vol. 176, no. 1, pp. 177–192, 2007.

[13] J. Gibbs, G. Kendall, and E. Özcan, “Scheduling english football fixtures over the holiday period using hyper-heuristics,” in *Parallel Problem Solving from Nature, PPSN XI.* Springer, 2010, pp. 496–505.

[14] P. Ross, S. Schulenburg, J. G. Marín-Blázquez, and E. Hart, “Hyper-heuristics: learning to combine simple heuristics in bin-packing problems.” in *GECCO*, 2002, pp. 942–948.

[15] H. Terashima-Marín, J. C. Ortiz-Bayliss, P. Ross, and M. Valenzuela-Rendón, “Using hyper-heuristics for the dynamic variable ordering in binary constraint satisfaction problems,” in *MICAI 2008: Advances in Artificial Intelligence.* Springer, 2008, pp. 407–417.

[16] P. Garrido and C. Castro, “Stable solving of cvrps using hyperheuristics,” in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation.* ACM, 2009, pp. 255–262.

[17] G. Ochoa, M. Hyde, T. Curtois, J. Vazquez-Rodriguez, J. Walker, M. Gendreau, G. Kendall, B. McCollum, A. Parkes, S. Petrovic, and E. Burke, *HyFlex: A Benchmark Framework for Cross-domain Heuristic Search*, ser. LNCS. Heidelberg: Springer, 2012, vol. 7245, pp. 136–147.

[18] M. Hyde, G. Ochoa, J. A. Vázquez-Rodríguez, and T. Curtois, "A hyflex module for the max-sat problem," *University of Nottingham*, 2011.

[19] M. Hyde, G. Ochoa, T. Curtois, and J. Vázquez-Rodríguez, "A hyflex module for the one dimensional bin-packing problem," *School of Computer Science, University of Nottingham*, 2009.

[20] T. Curtois, G. Ochoa, M. Hyde, and J. A. Vázquez-Rodríguez, "A hyflex module for the personnel scheduling problem," *School of Computer Science, University of Nottingham*, 2010.

[21] J. A. Vázquez-Rodrıguez, G. Ochoa, T. Curtois, and M. Hyde, "A hyflex module for the permutation flow shop problem," *School of Computer Science, University of Nottingham*, 2009.

[22] M. Bellmore and G. L. Nemhauser, "The traveling salesman problem: a survey," *Operations Research*, vol. 16, no. 3, pp. 538–558, 1968.

[23] P. Toth and D. Vigo, *The vehicle routing problem.* Society for Industrial and Applied Mathematics, 2001.

[24] Á. Fialho, "Adaptive operator selection for optimization," *PhD, Ecole Doctorale d?Informatique, Universite Paris-Sud, Paris*, 2010.

[25] K. Li, A. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 1, pp. 114–130, 2014.

[26] N. R. Sabar, M. Ayob, G. Kendall, and R. Qu, "A dynamic multiarmed bandit-gene expression programming hyper-heuristic for combinatorial optimization problems," *Cybernetics, IEEE Transactions on*, vol. 45, no. 2, pp. 217–228, 2015.

[27] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[28] V. Kuleshov and D. Precup, "Algorithms for multi-armed bandit problems," *arXiv preprint arXiv:1402.6028*, 2014.

[29] C. Hartland, N. Baskiotis, S. Gelly, M. Sebag, and O. Teytaud, "Change point detection and meta-bandits for online learning in dynamic environments," 2007.

[30] L. DaCosta, A. Fialho, M. Schoenauer, and M. Sebag, "Adaptive operator selection with dynamic multi-armed bandits," in *Proceedings of the 10th annual conference on Genetic and Evolutionary Computation*. ACM, 2008, pp. 913–920.

[31] R. A. Gonçalves, C. P. Almeida, and A. Pozo, "Upper confidence bound (ucb) algorithms for adaptive operator selection in moea/d," in *Evolutionary Multi-Criterion Optimization*. Springer, 2015, pp. 411–425.

[32] J. L. Núnez and A. Ceballos, "A general purpose hyper-heuristic based on ant colony optimization," 2011.

[33] K. Sim, "Ksats-hh: A simulated annealing hyper-heuristic with reinforcement learning and tabu-search," *Entry in the Cross-domain Heuristic Search Challenge available from http://www. asap. cs. nott. ac. uk/external/chesc2011/index. html, J une*, 2011.

[34] P.-C. Hsiao, T.-C. Chiang, and L.-C. Fu, "A variable neighborhood search-based hyperheuristic for cross-domain optimization problems in chesc 2011 competition," in *Fifty-Third Conference of OR Society (OR53), Nottingham, UK*, 2011.

[35] M. Misir, K. Verbeeck, P. De Causmaecker, and G. Vanden Berghe, "A new hyper-heuristic implementation in hyflex: a study on generality," in *Proceedings of the 5th*

Multidisciplinary International Scheduling Conference: Theory & Application, 2011, pp. 374–393.

[36] J. H. Drake, E. Özcan, and E. K. Burke, “An improved choice function heuristic selection for cross domain heuristic search,” in *Parallel Problem Solving from Nature-PPSN XII.* Springer, 2012, pp. 307–316.

[37] A. Kheiri and E. Keedwell, “A sequence-based selection hyper-heuristic utilising a hidden markov model,” in *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference.* ACM, 2015, pp. 417–424.

[38] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *Evolutionary Computation, IEEE Transactions on*, vol. 1, no. 1, pp. 67–82, 1997.

[39] P. Cowling, G. Kendall, and E. Soubeiga, “A hyperheuristic approach to scheduling a sales summit,” in *Practice and Theory of Automated Timetabling III.* Springer, 2001, pp. 176–190.

[40] P. Cowling, G. Kendall, and L. Han, “An investigation of a hyperheuristic genetic algorithm applied to a trainer scheduling problem,” in *Proceedings of the Congress on Evolutionary Computation 2002, CEC 2002*, 2002, pp. 1185–1190.

[41] P. Rattadilok, “An investigation and extension of a hyper-heuristic framework,” *Informatica*, vol. 34, no. 4, 2010.

[42] M. Maashi, E. Özcan, and G. Kendall, “A multi-objective hyper-heuristic based on choice function,” *Expert Systems with Applications*, vol. 41, no. 9, pp. 4475–4493, 2014.

[43] R. A. Gonçalves, J. N. Kuk, C. P. Almeida, and S. M. Venske, “Moea/d-hh: A hyper-heuristic for multi-objective problems,” in *Evolutionary Multi-Criterion Optimization.* Springer, 2015, pp. 94–108.

[44] L. Han, G. Kendall, and P. Cowling, "An adaptive length chromosome hyperheuristic genetic algorithm for a trainer scheduling problem," in *Proceedings of the 4th Asia-Pacific conference on simulated evolution and learning (SEAL02)*, 2002, pp. 267–271.

[45] L. Han and G. Kendall, "An investigation of a tabu assisted hyper-heuristic genetic algorithm," in *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, vol. 3. IEEE, 2003, pp. 2230–2237.

[46] G. Ochoa, R. Qu, and E. K. Burke, "Analyzing the landscape of a graph based hyper-heuristic for timetabling problems," in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation.* ACM, 2009, pp. 341–348.

[47] J. Thomas and N. S. Chaudhari, "Design of efficient packing system using genetic algorithm based on hyper heuristic approach," *Advances in Engineering Software*, vol. 73, pp. 45–52, 2014.

[48] E. Burke, G. Kendall, D. L. Silva, R. O'Brien, D. L, A. Silva, and E. Soubeiga, "An ant algorithm hyperheuristic for the project presentation scheduling problem," in *In: Proceedings of the Congress on Evolutionary Computation 2005 (CEC 20005). Volume 3.* IEEE press, 2005, pp. 2263–2270.

[49] P.-C. Chen, G. Kendall, and G. Berghe, "An ant based hyper-heuristic for the travelling tournament problem," in *Computational Intelligence in Scheduling, 2007. SCIS '07. IEEE Symposium on*, April 2007, pp. 19–26.

[50] M. Larose, "A hyper-heuristic for the chesc 2011," 2011.

[51] E. Burke, T. Curtois, M. Hyde, G. Kendall, G. Ochoa, S. Petrovic, J. Vázquez-Rodríguez, M. Gendreau *et al.*, "Iterated local search vs. hyper-heuristics: Towards general-purpose search algorithms," in *Evolutionary Computation (CEC), 2010 IEEE Congress on.* IEEE, 2010, pp. 1–8.

[52] M. Ayob and G. Kendall, "A monte carlo hyper-heuristic to optimise component placement sequencing for multi head placement machine," in *Proceedings of the international conference on intelligent technologies, InTech*, vol. 3, 2003, pp. 132–141.

[53] N. R. Sabar, M. Ayob, G. Kendall, and R. Qu, "Grammatical evolution hyper-heuristic for combinatorial optimization problems," *Evolutionary Computation, IEEE Transactions on*, vol. 17, no. 6, pp. 840–861, 2013.

[54] H. Robbins *et al.*, "Some aspects of the sequential design of experiments," *Bulletin of the American Mathematical Society*, vol. 58, no. 5, pp. 527–535, 1952.

[55] Á. Fialho, L. Da Costa, M. Schoenauer, and M. Sebag, "Extreme value based adaptive operator selection," in *Parallel Problem Solving from Nature–PPSN X*. Springer, 2008, pp. 175–184.

[56] M. Nawaz, E. E. Enscore, and I. Ham, "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem," *Omega*, vol. 11, no. 1, pp. 91–95, 1983.